SUPER sinc

n° 1 Giugno 1984
L. 3.500
Spedizione in abbonamento postale Gruppo III/70

1

Una pubblicazione della J.soft EDITRICE

SUPERSINC
la rivista per gli utenti dei prodotti SINCLAIR

BILL IL MURATORE

BOWLING



ATTACCO MISSILISTICO

OK CORRAL

UN'OCCHIATA AL QL

INFERNO

PERSONAL COMPUTER: il meglio, in edicola!

BIT
La più letta, la prima e più diffusa.
TEST - 64 Pagine di PROGRAMMI
- Inserto SPECIALE SCUOLA
- E tutte le NOVITA' HARDWARE E SOFTWARE

PERSONAL SOFTWARE
- L'unica che presenta software per tutti i personal: Commodore, Apple, Sinclair, T.I., HP, Sharp, ecc.



Bit e Personal Software sono riviste firmate
GRUPPO EDITORIALE JACKSON

compra il tuo Spectrum con la supergaranzia



La Rebit Computer, distributrice per l'Italia dei prodotti SINCLAIR, ha messo a punto una nuova **sùpergaranzia** che ti darà i seguenti vantaggi:

1° Prezzo ridotto nell'acquisto dell'interfaccia programmabile.

2° Tessera sconto sull'acquisto dei programmi.

3° Tariffa ridotta per l'abbonamento a "Sperimentare con il Computer"

4° Libro sulle interfacce e sui microdrives.

Un risparmio di oltre 70.000 lire.

NON PERDERE QUESTA OCCASIONE
al prezzo ECCEZIONALE
di L. 49.000 + IVA
anzichè
L. 99.000 + IVA

PLAYER ONE
PLAYER TWO
PROGRAMMABLE JOYSTICK INTERFACE

PROGRAMMABLE
JOYSTICK
INTERFACE
ZX Spectrum

TENKOLEK

ADD ACTION TO YOUR COMPUTER GAMES !!
TENKOLEK®

inoltre riceverete in OMAGGIO direttamente a casa, l'opuscolo in italiano :
SINCLAIR ZX Interfaccia 1
ZX Microdrive
del valore di L. 10.000

Edizione italiana
sinclair
ZX SPECTRUM
Microdrive and Interface 1 manual

REBIT COMPUTER
A DIVISION OF G.B.C.

sinclair

È vero: piccolo è bello!

Alla scoperta dello ZX SPECTRUM

a cura di **Rita Bonelli**



ZX Spectrum è l'ultimo nato della famiglia Sinclair. È un calcolatore a colori di piccole dimensioni, ma di grandissime possibilità. Imparare a usarlo bene può essere fonte di molte piacevoli scoperte. Questo libro vi aiuta a raggiungere lo scopo. In 35 brevi e facilissimi capitoli non solo imparerete tutto sulla programmazione in BASIC, ma arriverete anche a usare efficientemente il registratore e a sfruttare al meglio le stampe. Soprattutto capirete la differenza tra il vostro Spectrum e gli altri computer.

320 pagine. Lire 22.000 Codice 337 B

GRUPPO EDITORIALE JACKSON



Per ordinare il volume utilizzare l'apposito tagliando inserito in fondo alla rivista

Scrive, suona, gioca, entusiasma

Gaetano Marano

66 PROGRAMMI PER ZX 81 E ZX80 CON NUOVA ROM + HARDWARE



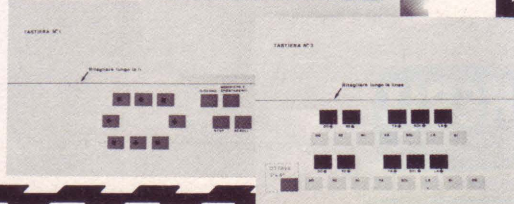
Per le sue qualità e il suo modestissimo prezzo lo ZX 81 della Sinclair è il computer più venduto nel mondo.
Oggi, sempre con una modestissima spesa, si può imparare a sfruttare questo eccezionale strumento al limite delle sue capacità. Basta scorrere questo libro per scoprire quante cose lo ZX 81 può fare con l'aggiunta di alcuni semplici ed economici componenti. Ad esempio, tramite un semplice circuito musicale può riprodurre 50 note su 4 ottave e, sempre grazie a una modifica hardware da poche migliaia di lire, lo ZX 81 diventa anche l'unico computer in grado di conferire effetti sonori ai giochi inseriti tra i suoi programmi. Ma non è tutto. Un'altra novità di quest'opera, preziosa anche per chi possiede lo ZX 80 con ROM, è il regalo di alcune tastiere disegnate da sovrapporre a quella sensitiva dell'apparecchio, per ricavarne altre, speciali funzioni.

136 pagine. Lire 12.000 Codice 520 D

Per ordinare il volume utilizzare l'apposito tagliando inserito in fondo alla rivista



GRUPPO EDITORIALE JACKSON



SUPER SINC

7 EDITORIALE POINT
QUALCOSA DI NUOVO!
di Pietro Dell'Orco

9 READ & WRITE
LA POSTA DEI LETTORI

10 OPEN
UN OCCHIATA AL QL
di Marcello Spero

14 LA CULTURA INFORMATICA IF ...then
PROGRAMMAZIONE STRUTTURATA
di Tim Hartnell
trad. di Marcello Spero

18 REM:HW
LE INTERRUZIONI, TEORIA E PRATICA SULLO ZX SPECTRUM
di Marcello Spero

23 LOAD
ATTACCO MISSILISTICO
di Wolfgang Huebl
trad. e adatt. di Ivan Cerè

26 IPPODROMO
di N.C. Pearson
trad. e adatt. di Ivan Cerè

29 BILL IL MURATORE
di C. Elliston
trad. e adatt. di Ivan Cerè

32 BOWLING
di Andrew Cook
trad. e adatt. di Ivan Cerè

35 OK CORRAL
di R. Page
trad. e adatt. di Ivan Cerè

38 INFERNO
di Akram Malik
trad. e adatt. di Ivan Cerè

42 PLOT
di Carlo Panzalis

49 KING KONG
di Nicholas Wyre
trad. e adatt.
di Marcello Spero

54 CHOMP CHOMP
di Marcello Spero

58 R·U·N
RECENSIONI SOFTWARE
a cura
di Lucio Bragagnolo

60 PRINT
RECENSIONI LIBRI
a cura
di Lucio Bragagnolo

63 ZX CLUB
I CLUB SINCLAIR

65 input·output
PICCOLI ANNUNCI

J.soft s.r.l
DIREZIONE, REDAZIONE, AMMINISTRAZIONE
Via Rosellini, 12
20124 MILANO
Tel. (02) 6888228

DIRETTORE RESPONSABILE:
Pietro Dell'Orco

COORDINAMENTO TECNICO:
Riccardo Paolillo

REDAZIONE
Lucio Bragagnolo
Marcello Spero

HANNO COLLABORATO A QUESTO NUMERO
C. Panzalis, I. Cerè

GRAFICA E IMPAGINAZIONE
Luigi Chiesa
Raffaella Toffolatti

FOTOCOMPOSIZIONE:
d&b Via Vignola, 5
Tel. 02/59.85.08
20133 MILANO

CONTABILITÀ:
Giulia Pedrazzini
Flavia Bonaiti

AUTORIZZAZIONE ALLA PUBBLICAZIONE:
Tribunale di Milano n° 199
del 14.04.1984

STAMPA:
Litografia del Sole
Albairate (MI)

PUBBLICITÀ
Concessionario per l'Italia e l'Estero
Reina s.r.l.
Via Washington, 50
20046 MILANO
Tel. (02) 4988066 (5 linee R.A.)
Tlx. 316213 REINA I

Concessionario esclusivo per la
DIFFUSIONE in Italia e Estero:
SODIP - Via Zuretti, 25
20125 MILANO

Spedizione in abbonamento postale Gruppo III/70
Prezzo della rivista L. 3.500
Numero arretrato L. 6.000
Abbonamento annuo (11 numeri) L. 32.000; per l'estero L. 50.000
i versamenti devono essere indirizzati a:
TechnoClub s.r.l.
Via Rosellini, 12
20124 MILANO
mediante emissione di assegno bancario, cartolina, vaglia o utilizzando il c/c postale n. 19445204
Per i cambi di indirizzo indicare, oltre al nuovo, anche l'indirizzo precedente ed allegare alla comunicazione l'importo di L. 500, anche in francobolli

© TUTTI I DIRITTI DI RIPRODUZIONE O TRADUZIONE DEGLI ARTICOLI PUBBLICATI SONO RISERVATI

GLI ARTICOLI TRADOTTI SONO TRATTI DALLA RIVISTA
ZX COMPUTING - ARGUS SPECIALIST PUBLICATION L.T.D.
1 GOLDEN SQUARE LONDON
W1R3AB

Per collaborare a SUPERSINC

La maggior parte dei seguenti suggerimenti ha lo scopo di migliorare l'accuratezza e la velocità di pubblicazione di un articolo; rispettando questi consigli si accresceranno le probabilità che un vostro lavoro venga pubblicato. La rivista è interessata ad articoli e programmi riguardanti la linea Sinclair. Siamo più interessati al contenuto di un articolo piuttosto che al suo stile, e soprattutto gli articoli devono essere chiari ed esaurienti. La seguente guida permetterà che le vostre buone idee e i vostri programmi vengano più facilmente accettati per la pubblicazione:

1 l'angolo superiore sinistro della prima pagina dovrà contenere: nome, cognome, indirizzo, numero telefonico, codice fiscale e data di spedizione.

2 l'angolo superiore destro della prima pagina dovrà contenere il nome del computer al quale il lavoro si riferisce, unitamente alla configurazione richiesta (memoria occorrente, eventuali periferiche e così via).

3 il titolo sottolineato dell'articolo dovrà iniziare a circa due terzi in altezza della prima pagina.

4 le pagine seguenti potranno essere battute normalmente, con la condizione che l'angolo superiore destro contenga un'abbreviazione del titolo e del cognome, unitamente al numero di pagina progressivo. Per esempio, Horace Goes.../Brambilla/2.

5 tutte le linee del testo dell'articolo dovranno essere battute con spazio 2 o spazio 3, e un margine di circa un centimetro dovrà trovarsi ad entrambi i lati dello scritto.

6 dovrà essere usata una carta formato A4 e lo scritto dovrà occupare un solo lato del foglio (caratteri maiuscoli e minuscoli).

7 i fogli dovranno essere uniti con una clip.

8 avendo intenzione di spedire più di un articolo, questi dovranno essere inviati separatamente insieme alla rispettiva copia su supporto magnetico.

9 programmi brevi (meno di 20 linee) potranno essere inseriti nel testo, mentre programmi più lunghi dovranno essere listati separatamente. È **ESSENZIALE** per noi disporre di una copia del programma registrata più volte su supporto magnetico, su entrambi i lati dello stesso. È preferibile usare nastri di buona qualità e di lunghezza non eccessiva; la cassetta o la cartuccia per Microdrive dovranno essere etichettati con il nome dell'autore, il titolo dell'articolo, il computer interessato e soprattutto le eventuali espansioni richieste. Come suggerimenti di programmazione, si consiglia di usare, per esigenze di stampa listati, le istruzioni INK, PAPER, INVERSE piuttosto che scrivere direttamente in INVERSE VIDEO. Un rapido controllo dei programmi per operare queste sostituzioni sarà da noi estremamente apprezzato.

10 per maggior chiarezza, all'interno dell'articolo è conveniente usare caratteri maiuscoli riferendosi a istruzioni BASIC (esempio RETURN, LIST, RND, PRINT etc.). Se si desidera evidenziare una parola, è preferibile sottolinearla piuttosto che scriverla in carattere maiuscolo.

11 gli articoli ed i programmi potranno avere qualsiasi lunghezza — da una routine di una sola linea fino a programmi molto complessi.

12 volendo includere fotografie, queste dovranno essere in formato 24 × 36, o 6 × 6, in bianco e nero o diapositive.

13 non prenderemo in considerazione articoli che siano stati sottoposti ad altre case editrici.

14 il compenso per la collaborazione prestata sarà commisurato alla complessità e all'interesse del programma (da un minimo di L. **50.000** a un massimo di L. **300.000**). Il pagamento è effettuato in caso di pubblicazione del lavoro.

15 il materiale ricevuto e non pubblicato non verrà restituito.

Spedite i vostri lavori a:

SUPERSINC
Via Rosellini, 12
20124 Milano

e saremo lietissimi di pubblicare i contributi migliori.

La Redazione

POINT

Qualcosa di nuovo!

Eccoci qua! Ci aspettavate? Speriamo di sı! Gli utenti a cui ci rivolgiamo normalmente sono alla loro prima esperienza in questo campo.

Questo determina, oltre ad un grande entusiasmo, l'impatto immediato con la realtà che si è creata attorno a questo tipo di macchine.

L'utente che avvicina per la prima volta un personal computer, dopo aver avidamente letto e provato quei piccoli e volutamente banali programmi proposti dal manuale, si ritrova a dover cercare, su tutte le riviste di settore, altro materiale da far "girare" sul nuovo amico.

La ricerca cui si accennava è talvolta vana e spesso economicamente onerosa.

Ecco dunque **SUPERSINC** nato proprio per risolvere, speriamo in maniera brillante, il problema.

SUPERSINC è infatti una rivista dedicata ad un computer specifico e tutti gli argomenti trattati (hardware, software, libri, notizie, etc.) saranno rigorosamente legati a quella macchina o famiglia di macchine.

La rivista, oltre a porsi come naturale complemento all'utilizzo del computer, cercherà di fornire all'utente notizie e informazioni relative alle novità del mercato hardware e software.

I programmi proposti sono selezionati e tradotti dalle migliori riviste inglesi e americane.

SUPERSINC è una rivista indipendente: non legata, cioé, al costruttore o al distributore del computer trattato. Questa scelta garantisce al lettore l'obbiettività necessaria di una rivista che nasce proprio nell'interesse dell'utente.

Inutile dire che auspichiamo la collaborazione dei lettori che, come sempre, sono i giudici assoluti di quanto è proposto; attendiamo dunque giudizi, commenti e quant'altro possa esserci utile per fare la rivista su misura per voi.

Pietro Dell'Orco

OLTRE L'ORIZZONTE CON LO SPECTRUM

77 PROGRAMMI PER SPECTRUM

GRAFICA - BUSINESS GRAFICA - UTILITY - ANIMAZIONI - MUSICA - GIOCHI





di Gaetano Marano

GRUPPO EDITORIALE JACKSON

77 PROGRAMMI PER SPECTRUM

150 Pagine. **30** illustrazioni a colori
Cod. **555 A**
L. 16000



GRUPPO EDITORIALE JACKSON

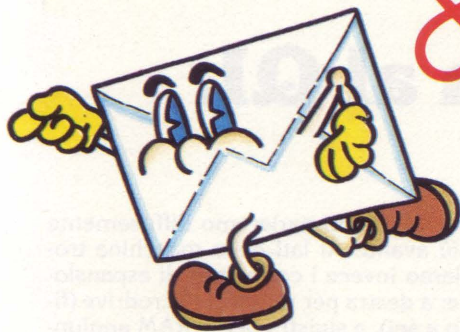
E PER LO ZX81...

66 PROGRAMMI PER ZX81 E ZX80 CON NUOVA ROM + HARDWARE

144 Pagine
Cod. **520 D**
L. **12000**



READ &WRITE



La posta dei lettori

Cari lettori,
come potete immaginare è un po' difficile, nel primo numero di una rivista, disporre di lettere che non siano riprese da altri giornali o addirittura inventate di sana pianta: così, solo per questa volta, ci permettiamo di invadere lo spazio che dai prossimi numeri verrà occupato da voi. Tutto ciò che abbiamo da dirvi è che... aspettiamo i vostri contributi! E un'altra cosa ancora.
Ci piacerebbe moltissimo vedere questa rubrica diventare, più che un botta-e-risposta fra voi e **SUPERSINC**, uno spazio dove scambiarsi esperienze, trucchi, consigli, fare domande e dare risposte a tutto ciò che può essere di interesse generale. Avete trovato la soluzione di un'adventure? Avete scoperto cosa accade digitando PLOT 127,88:DRAW -50,50,2↑14? Siete in difficoltà con i vostri Microdrive? La linea x del programma tot provoca la fusione dell'alimentatore? In parole povere, scriveteci per qualunque cosa abbiate da dirci, e pubblicheremo le lettere più interessanti e divertenti. Nell'attesa, vi diamo qualche esempio di "lettere a **SUPERSINC**", naturalmente inventate e copiate con la massima spudoratezza. A risentirci!

La redazione

Ho scoperto, durante una partita a Manic Miner, un fantastico "bug", che consente nientemeno di cominciare il gioco partendo da una qualsiasi dei 20 schermi a scelta. Consiste nel digitare (a questo punto ci fermiamo per non disgustare i puristi di Manic Miner. Ma se i non puristi sono tanti, potremmo anche dirvelo... una volta verificata la cosa. n.d.r.).

Ben Hordyment - Clapham, London SW4

Caro SUPERSINC,
nel vostro gioco "Pippo, Pluto e Paperino", sostituendo la linea 450 con LET A$ = "GAMBADILEGNO", la routine di scrolling video raddoppia la velocità di esecuzione.

Walter. E. Disney - Epcot (USA)

Cara Redazione,
ho scoperto un sistema alternativo al BREAK per fermare un programma: occorre premere contemporaneamente i tasti Z, CAPS SHIFT e SYMBOL SHIFT. Trovo che potrebbe tornare utile nel caso non si voglia usare il tasto BREAK.

Peter Ryan - Middlesex

Caro SUPERSINC,
accludo un miniprogramma per tracciare spirali:

```
5 INPUT x,y
10 INPUT "Che numero?";p
15 PLOT 128,88
20 FOR a=0 TO 500*PI STEP PI/p
30 DRAW x*a*SIN a, y*a*COS a
40 NEXT a
```

Provate anche ad aggiungere OVER 1.

Simon White - Gloucestershire

Spettabile redazione,
ho letto della prossima uscita di un nuovo computer Sinclair. Non potreste darmi delle informazione più dettagliate in proposito? La Spectrum continuerà a essere prodotto o farà la fine del TI/99?

Ruggero Amiconi - Abbiategrasso

Caro SUPERSINC,
sono un ragazzo di 11 anni appassionato di computer. Mi piacerebbe sapere dove posso trovare i Microdrive per il mio Spectrum e se potrò copiarci sopra i miei videogames.

Alessio Ronconi - Milano

Egregio Signor Direttore,
di ritorno da un viaggio in Inghilterra ho portato a casa in regalo per mio figlio la cassetta Atic Atac.
Sono ormai due settimane che stiamo cercando la soluzione del gioco, senza trovarla. Lei non sa se qualcuno ci potrebbe aiutare o dare dei consigli? La ringrazio vivamente.

Salvatore Aguglia - Benevento

Spettabile Redazione di SUPERSINC,
sono uno studente diciassettenne da circa otto mesi possessore di uno Spectrum 48K. Con l'esercizio ho sviluppato una, credo, notevole capacità di programmazione e da qualche tempo ho cominciato a scrivere dei videogiochi per me e per i miei amici. È possibile farli pubblicare su SUPERSINC o riuscirli a venderli? Vi allego una cassetta C90 con alcune delle mie "creazioni". Ringraziandovi per l'attenzione vi saluto.

Edoardo Cecchi - Firenze

Egregio Direttore,
si può collegare lo Spectrum con un Commodore 64? È da tempo che io e un mio collega di ufficio (che ha il suddetto Commodore) cerchiamo un sistema rapido per scambiarci programmi senza dover ricopiare i reciproci listati, ma, forse per la nostra relativa inesperienza di programmazione, non siamo riusciti finora ad ottenere alcun risultato. Esiste qualche programma per realizzare ciò o il nostro è un caso senza speranza?

Giovanni Mengoni - Castelfidardo (AN)

Gentile redazione di SUPERSINC,
ho comprato il gioco "ah diddums" per il mio Spectrum, solo che non riesco a copiarlo. Come posso effettuare la copiatura, al solo scopo (glielo garantisco) di avere una copia di sicurezza nel caso che quella che ho comprato si rovini?

Francesco De Curtis - Castellammare di Stabia (NA)

Potete consigliarmi dei libri che spieghino dettagliatamente il linguaggio macchina dello Spectrum?

Stefano Bonora - Bologna

Carissima Redazione di SUPERSINC,
ho appena comprato uno Spectrum 48K per la mia attività di programmazione "hobbistica", e ho letto che su quest'ultimo è possibile programmare anche in LISP e Pascal. Sapete se i rispettivi compilatori sono disponibili in Italia o se bisogna per forza rivolgersi all'estero?

Caterina Donati - Pisa



Un'occhiata al QL

di **Marcello Spero**

Passata la prima ondata di entusiasmo esaminiamo caratteristiche, pregi e difetti dell'ultimo nato di casa Sinclair.

Lo scorso gennaio è stato presentato, con un'uscita a sopresa nella migliore tradizione Sinclair, il nuovo **QL**. Promettente nel nome ("Quantum Leap", salto qualitativo) ed indubbiamente allettante nelle caratteristiche, questa macchina ha sorprendentemente sollevato parecchie perplessità.
Sir Clive Sinclair, d'altra parte, è certo di aver aperto un nuovo segmento nel mercato dell'informatica individuale, finora tradizionalmente suddiviso in home, personal e business.
Il **QL**, infatti, se per prezzo può senz'altro essere annoverato fra gli home computer, quanto a prestazioni è al vertice della fascia business.
Questa sua doppia natura è confermata da una parte dalla presenza di due prese per joystick, dall'altra dalla notevole dotazione di software professionale compresa nel prezzo.
Questa, come vedrete, è solo una delle molte, ed in un certo senso affascinanti, contraddizioni del **QL**, un computer che forse non siamo ancora preparati a valutare correttamente.

L'esterno

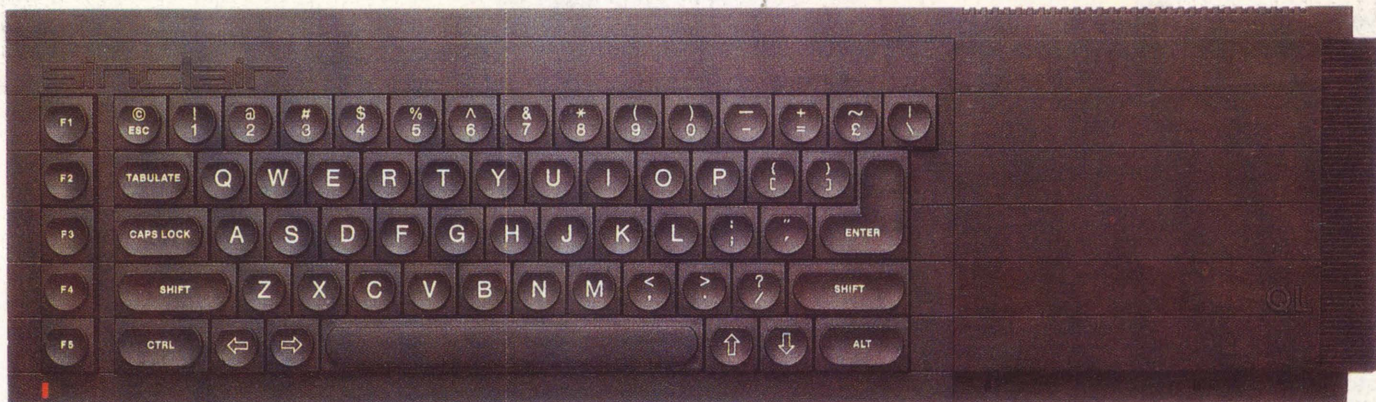
Molto elegante nella sua essenzialità il **QL** si presenta come un oggetto nero, dai riflessi quasi metallici, largo poco meno di 50 cm, profondo circa 14 cm e con uno sviluppo in altezza di non più di qualche centimetro, leggermente inclinato in avanti per l'effetto di una coppia di piedini smontabili.
La tastiera, che occupa la superficie dell'apparecchio eccettuata una porzione quadrata sulla destra, è standard per dimensioni (finalmente!) e disposizione dei tasti (quasi ...), che sono ben 65, fra cui cinque tasti di funzione programmabili. Gli esemplari che abbiamo avuto modo di vedere, essendo destinati al mercato britannico, montavano il tipo **QWERTY**, ma è prevista per le unità destinate al nostro mercato la tastiera nazionalizzata.
Rinviando un giudizio definitivo alla vera prova della macchina, che speriamo di poter effettuare quanto prima, possiamo dire che la prima impressione suscitata da questa tastiera è senz'altro positiva. I tasti, costruiti ad intarsio per garantirne la durata della leggibilità, producono un "click" chiaramente udibile. Sotto di loro, inoltre è presente una membrana in gomma, a protezione dei contatti elettrici. Nel complesso, insomma, si tratta di una tastiera che pur non raggiungendo il livello di eccellenza di alcuni prodotti, d'altronde ben più costosi, presenti sul mercato, dovrebbe soddisfare anche l'utenza più esigente. L'unico appunto potrebbe riguardare la posizione del tasto **CTRL** e di alcuni simboli, che avrebbe potuto essere più rigorosamente standard. Infine una pignoleria: non sarebbe stato proprio possibile collocare in quel quadrato vuoto un bel tastierino numerico?
Subito sotto il bordo anteriore, in corrispondenza dell'area quadrata qui accennavamo, troviamo le due feritorie dei Microdrive, la memoria di massa del **QL** di cui parleremo diffusamente più avanti. Ai lati della macchina troviamo invece i connettori di espansione: a destra per ulteriori Microdrive (fino a sei), a sinistra per la RAM aggiuntiva (0,5 Mbyte). Accanto al connettore per i Microdrive è inoltre presente un pulsantino di reset, per evitare, anche in casi estremi, la manovra di spegnimento e riaccensione, altrimenti necessaria per ripristinare le condizioni iniziali. La scelta di questa posizione dovrebbe garantire l'utilizzatore da reset accidentali, con conseguenze sempre disastrose.
Il pannello posteriore ospita invece, oltre alle prese per monitor e televisore, utilizzabili contemporaneamente, ed all'ingresso dell'alimentazione, le uscite per la comunicazione con l'esterno, che comprendono due porte per il collegamento in rete locale, due interfacce seriali tipo RS-232-C, due prese per joystick. È inoltre presente un connettore per cartucce ROM, mentre non esiste, sempre nella tradizione Sinclair, un interruttore di accensione.

L'interno

Molto ordinato si presenta suddiviso in tre grandi zone: un'area centrale occupata dalla piastra del circuito vero e proprio, una porzione posta anteriormente sulla destra contenente la mec-



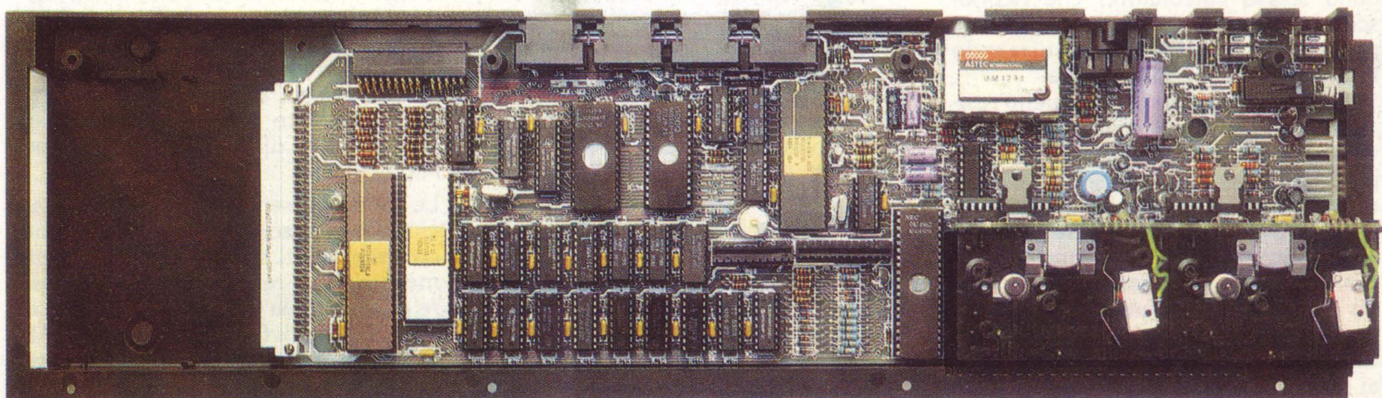
In questa foto e nelle seguenti, immagini del Sinclair QL.



canica dei due Microdrive ed infine un'ampia zona vuota sulla sinistra destinata ad accogliere l'espansione di memoria.
Fra i numerosi circuiti integrati spiccano i due processori affiancati dalla lunga fila di memorie, ed il modulatore per l'uscita TV, dello stesso tipo utilizzato nello Spectrum. La mancanza di qualsiasi tipo di documentazione ci impedisce di indagare più a fondo sul significato dei numerosi integrati minori sparsi un po' dappertutto.
Vediamo invece in dettaglio l'architettura del sistema, fonte anch'essa di polemiche sulla validità di alcune scelte. La CPU, cioè il cuore operativo del sistema, è costituita da un processore 68008 Motorola di cui parleremo diffusamente più avanti, data l'importanza di questa scelta. Ad esso sono affiancati, per gestire i rapporti con l'esterno, un processore ausiliario Intel 8049 e due circuiti integrati appositamente progettati realizzati per il **QL**. In questo modo la CPU è sgravata da tutte le attività che potrebbero rallentarla. In particolare, il processore ausiliario si occupa della tastiera, della generazione dei suoni e della ricezione attraverso le due interfacce RS-232-C, mentre i due integrati controllano l'uno il video e la memoria e l'altro i Microdrive, l'orologio (il **QL** contiene un orologio in tempo reale indipendente), il collegamento in rete locale, nonché la trasmissione di dati attraverso le interfacce RS-232.
Per quanto riguarda la memoria troviamo una ROM da 32 Kbyte contenente linguaggio e sistema operativo, e 128 Kbyte di RAM. In un futuro molto prossimo, inoltre, sarà disponibile un'espansione di memoria da ben mezzo megabyte che, inserita nell'alloggiamento già previsto, porterà la memoria totale del sistema al rispettabile livello di 640 Kbyte.
Una tale disponibilità di memoria, anche se verrà sfruttata in pratica da una fetta molto ristretta di utenza, dà un idea delle possibilità offerte dall'adozione del 68008 come CPU. Per questo, e per la novità costituita dall'impiego di un processore di tale potenza in una macchina di questa categoria, esaminiamo più in dettaglio la CPU del **QL**.

La CPU

È costituita, come abbiamo detto, da un processore 68008 Motorola. La serie contrassegnata dalla sigla 68000, di cui il 68008 fa parte, sta riscuotendo un notevole successo, che porterà probabilmente questo tipo di processore a divenire di uso standard nell'ambito delle applicazioni personali dell'ultima generazione. Tutti i componenti della famiglia 68000 possiedono un'organizzazione interna a 32 bit, ma differiscono fra loro per l'ampiezza del bus dati, che è il veicolo con cui scambiano informazioni con l'esterno. Il 68000, infatti, possiede un bus a 16 bit, il futuro 68020 lo avrà a 32 bit mentre il 68008 può disporre di un bus a 8 bit, analogo cioè a quello utilizzato dal Z80 che equipaggia lo Spectrum.
Questo non vuole certamente dire che quando la Sinclair afferma di aver realizzato un computer con processore a 32 bit stia sostenendo qualcosa di non esatto, più semplicemente un computer di questo tipo, sebbene incomparabilmente più veloce di qualunque suoi concorrente provvisto di un processore a 8 bit, risulterà leggermente svantaggiato nelle operazioni I/O e di accesso alla memoria rispetto a quegli elaboratori che dispongono di processori con bus più ampi.
La scelta di questo tipo di processore, d'altra parte, è giustificata dal fatto che organizzare la memoria intorno ad un bus ad 8 bit è, allo stato attuale delle cose, più semplice e più economico.
Possedere un'architettura interna a 32 bit, comunque, conferisce a tutto il sistema una grande potenza. Il 68008, infatti, possiede ben 17 registri di 32 bit ciascuno e questo, come tutti coloro che programmano in linguaggio macchina sanno bene, vuol dire dover usare un numero inferiore di istruzioni per eseguire le stesse operazioni.
Per quanto riguarda la quantità di memoria direttamente indirizzabile, invece, il 68008, al pari degli altri componenti della famiglia, possedendo un bus indirizzi da 24 bit può gestire 16 Mbyte, quantità più che sufficiente per applicazioni personali (i processori a 8 bit, come lo Z80 od il 6502, possono indirizzare direttamente solo 64 Kbyte dovendo ricorrere, per gestire quantità maggiori, a procedimenti elaborati e lenti).
La maggior differenza fra i 68000 e i



vecchi 8 bit è comunque nel set di istruzioni in linguaggio macchina. Senz'altro non è necessario conoscere il linguaggio macchina per poter utilizzare un elaboratore, ma un numero sempre maggiore di persone si sta interessando a questo tipo di linguaggio. Del resto, gli stessi programmi BASIC vengono tradotti in linguaggio macchina durante l'esecuzione e quindi la qualità del linguaggio stesso condiziona di fatto l'efficienza di tutto il sistema.
In questo caso abbiamo un set composto da sole 56 istruzioni, per altro modificabili in molti modi per coprire ogni esigenza. Un numero tanto piccolo di istruzioni tanto potenti è facile da imparare e da ricordare, e rende i sistemi basati su questo tipo di processori da un lato molto efficienti nell'esecuzione dei programmi scritti in linguaggio ad alto livello, e dall'altro supporti ideali per la programmazione in linguaggio macchina o, meglio, in Assembly.

Sistema operativo e linguaggio

Sono entrambi contenuti nella ROM da 32 Kbyte. Il sistema operativo è assolutamente nuovo e creato appositamente per il **QL**, e si chiama appunto QDOS. La sua caratteristica più importante e, sotto molti punti di vista, eccezionale, è di essere un sistema monoutente multitasking. L'utilizzatore può cioè avere più programmi contemporaneamente in esecuzione e i cui risultati possono, se necessario, essere visualizzati in finestre sullo schermo, indipendenti l'una dall'altra. Questo è possibile grazie al QDOS, che gestisce opportunamente il tempo della CPU suddividendolo fra i vari programmi in esecuzione, che sono in tal modo eseguiti parallelamente.
Altra caratteristica di rilievo di questo sistema operativo è il sistema di ingressi/uscite, completamente indipendente dal tipo di periferiche collegate. Questo significa, in altre parole, che i programmi potranno essere scritti senza fare riferimento alla particolare unità esterna con cui dovranno essere utilizzati. Questa potrà venire specificata di volta in volta al momento dell'uso o, in alternativa, rilevata nel tipo e nelle modalità di trasmissione dallo stesso calcolatore, che riconoscerà automaticamente la particolare periferica collegata, organizzando l'input/output di conseguenza.
Vediamo ora il linguaggio. Quello residente è il BASIC, in una particolare versione potenziata denominata SuperBASIC, che pur ricalcando il BASIC dello Spectrum ne risulta nettamente superiore per il maggior numero di istruzioni, che gli conferiscono caratteristiche nuove. Il SuperBASIC è infatti un linguaggio strutturato, cioè in possesso di tutte quelle strutture (IF THEN ELSE, REPEAT, UNTIL, ecc.), comuni ai linguaggi di più recente concezione quali Pascal, ADA, ecc., che consentono la stesura di programmi strutturati a blocchi e praticamente privi di istruzioni GO TO, con grande vantaggio per la loro comprensione. Altra fondamentale novità è la sua espandibilità, per mezzo di istruzioni DEF PROCEDURE, infatti, è possibile creare procedure che saranno richiamate da un nome scelto da noi, che funzioneranno perciò né più né meno allo stesso modo delle istruzioni presenti in ROM. È stato infine superato un problema presente nel BASIC tradizionale, e cioè la sua dipendenza, nella velocità di esecuzione, dalla lunghezza del programma.
Altre istruzioni che rendono più flessibile e potente la programmazione comprendono AUTO, per la numerazione automatica delle linee, BAUD per definire da BASIC la velocità di trasmissione seriale WINDOW e CURSOR per creare finestre video e posizionare al loro interno il cursore, EXEC per caricare sequenze di programmi ed eseguirle in parallelo, DATE$ per ottenere data e ora dall'orologio interno, e molte altre. Il medesimo SuperBASIC, infine, viene utilizzato come linguaggio di comando per il sistema operativo, rendendo veramente semplice la gestione del QDOS.
Due sono i modi previsti per la grafica ad alta dissoluzione: 512 × 256 pixel in quattro colori, e 256 × 256 in otto colori. Sono presenti inoltre una istruzione di scala, SCALE, ed una, BLOCK, per riempire con un determinato colore aree chiuse.
Il testo viene invece normalmente visualizzato in un formato di 85 colonne per 25 righe, che viene ridotto, nel caso si usi la TV, a 40-60 colonne. Il formato dei caratteri, e quindi il numero di colonne visualizzate, è comunque variabile da BASIC con l'istruzione CSIZE. La sottolineatura dei caratteri avviene ora in modo semplice con l'istruzione UNDER.



Visibile, in primo piano, l'espansione di memoria da 0,5 Mbyte.

Periferiche

La memoria di massa standard del **QL** è costituita dai Microdrive, che rappresentano un'evoluzione rispetto a quelli dello Spectrum. La loro capacità è infatti di 100 Kbyte, la velocità media di accesso 3,5 secondi, mentre quella di trasferimento raggiunge i 15 Kbyte al secondo. Con le due unità incorporate l'utente dispone perciò di 200 Kbyte in linea, mentre la configurazione massima di 6 Microdrive aggiuntivi porta la memoria di massa a 800 Kbyte. È inoltre previsto, in un futuro abbastanza prossimo, un'interfaccia per disco rigido tipo winchester.
A questo punto corre l'obbligo di dire che, per quanto non abbiano ancora in mano i dati relativi ad un uso reale dei Microdrive del **QL**, l'applicazione di questo tipo di memoria di massa ad un prodotto indirizzato al settore professionale ci lascia alquanto dubbiosi. L'unico vero argomento a loro favore, rispetto ad un sistema convenzionale a floppy disk è infatti il prezzo dei drive. Contro abbiamo la struttura seriale delle cartucce, che significa tempi effettivi di accesso che possono arrivare facilmente a 7 secondi, cioè più della maggior parte dei drive da 5 1/4. Inoltre avendo questi una capacità inferiore di circa 1/3 ed un prezzo più o meno doppio rispetto ai floppy da 5 e 1/4, il costo per l'immagazzinamento della medesima quantità di dati viene ad essere, con i Microdrive, circa sei volte superiore. Ciò non toglie che per un utilizzatore che non necessiti di grandi masse di dati e dia invece importanza all'ingombro del sistema il Microdrive possano rappresentare la soluzione ideale. Con l'affiancarsi del disco rigido, inoltre, si renderà disponibile ai grandi utenti un'unità adatta ad essere utilizzata anche in unione all'espansione di memoria, altrimenti difficilmente gestibile visto che può contenere l'equivalente di cinque cartucce per Microdrive.

Le uscite presenti sul retro della macchina le consentono il collegamento con stampanti seriali, nonché l'inserimento in una rete locale che può essere composta da **QL** e Spectrum fino ad un massimo di 64, denominata QLAN. Per il trasferimento dei dati attraverso questa rete la Sinclair dichiara una ve-

locità di 100 KBaud che, fatti i conti, risulta un po' superiore a quella prevista per la Local Area Network dello Spectrum, data per 5 KByte/sec. È realmente possibile collegare degli Spectrum a questa velocità?
Le periferiche aggiuntive già previste per il futuro comprendono, oltre l'interfaccia per il disco fisso e l'espansione di memoria, un'interfaccia parallela (chissà perché non è stata incorporata nella macchina) che includerà, pare, un generatore sonoro polifonico, un modem, un'interfacccia IEEE-488 (si tratta dell'interfaccia standard per il collegamento di strumentazione), un convertitore analogico/digitale. Non è invece compresa né prevista un'interfaccia per il collegamento con registratori a cassette, vista la presenza dei due Microdrive e la categoria di utenti cui la macchina si rivolge.



Clive Sinclair alla presentazione del «suo» nuovo QL.

Software

Ed eccoci finalmente al pezzo forte del **QL**. Il software fornito con la macchina, tutto di produzione Psion, è infatti veramente notevole, e costituisce una potente attrattiva al suo acquisto.
L'acquirente di un **QL** avrà a disposizione quattro programmi: **QL** Quill per l'elaborazione di testi, **QL** Abacus per calcoli (tipo tabellone Visicalc), **QL** Archive per l'archiviazione dati (database) ed infine **QL** Easel per realizzare qualsiasi tipo di grafici commerciali.
Estremamente sofisticati, omogenei nella concezione e perciò semplici nell'uso, con estese facilità di help richiamabili in qualsiasi momento, dovrebbero rappresentare quanto di meglio offre oggi il mercato in questo campo.
Vediamone qualche esempio. Il tabellone di calcolo è strutturato in 256 righe e 64 colonne, per un totale di 16.000 celle, e lavora con una precisione di 16 cifre significative. I riferimenti alle singole celle possono essere fatti con nomi definiti dall'utilizzatore. Il database consente l'uso di campi e record di lunghezza variabile. E, dulcis in fundo, i 4 package possono scambiarsi i dati l'uno con l'altro, costituendo così un unico sistema integrato.
Nonostante la grande facilità d'uso e l'ampia documentazione, sia sul manuale che negli stessi programmi, è stato istituito in Gran Bretagna (e speriamo avvenga lo stesso a suo tempo in Italia) il QLUB, con l'intento di risolvere qualsiasi problema dovesse presentarsi nell'uso del software come della macchina. Sarà cura del QLUB, inoltre, tenere i propri soci aggiornati inviando loro le nuove versioni dei package.
Le prospettive per l'immediato futuro sono altrettanto interessanti, e comprendono un compilatore Pascal, un'assembler 68000, software per l'emulazione di terminale e persino un compilatore C (a quando la compatibilità con UNIX?). Sono inoltre previste le versioni su cartuccia ROM, da inserire nell'apposito connettore, dei quattro package applicativi venduti con la macchina.

Conclusioni

Potente nell'hardware, ma soprattutto nel software, come macchine ben più costose, siamo certi che questa "macchinetta" da 400 sterline (poco più di un milione) farà molto parlare di sé. Con tastiera e programmi italiani come è nelle intenzioni della Rebit, importatore ufficiale, non avrà difficoltà ad imporsi. Siamo probabilmente di fronte ad una svolta nel modo di concepire il calcolatore personale. Potrà contrastare il passo a nomi illustri? Certo avrebbe il passo più sicuro se fosse maggiormente standardizzato. Una macchina con tastiera parzialmente non standard, sistema operativo non standard, linguaggio non standard e memoria di massa non standard potrà infatti suscitare qualche perplessità. Riuscirà il nuovo standard Sinclair ad imporsi? Solo il tempo potrà risponderci.



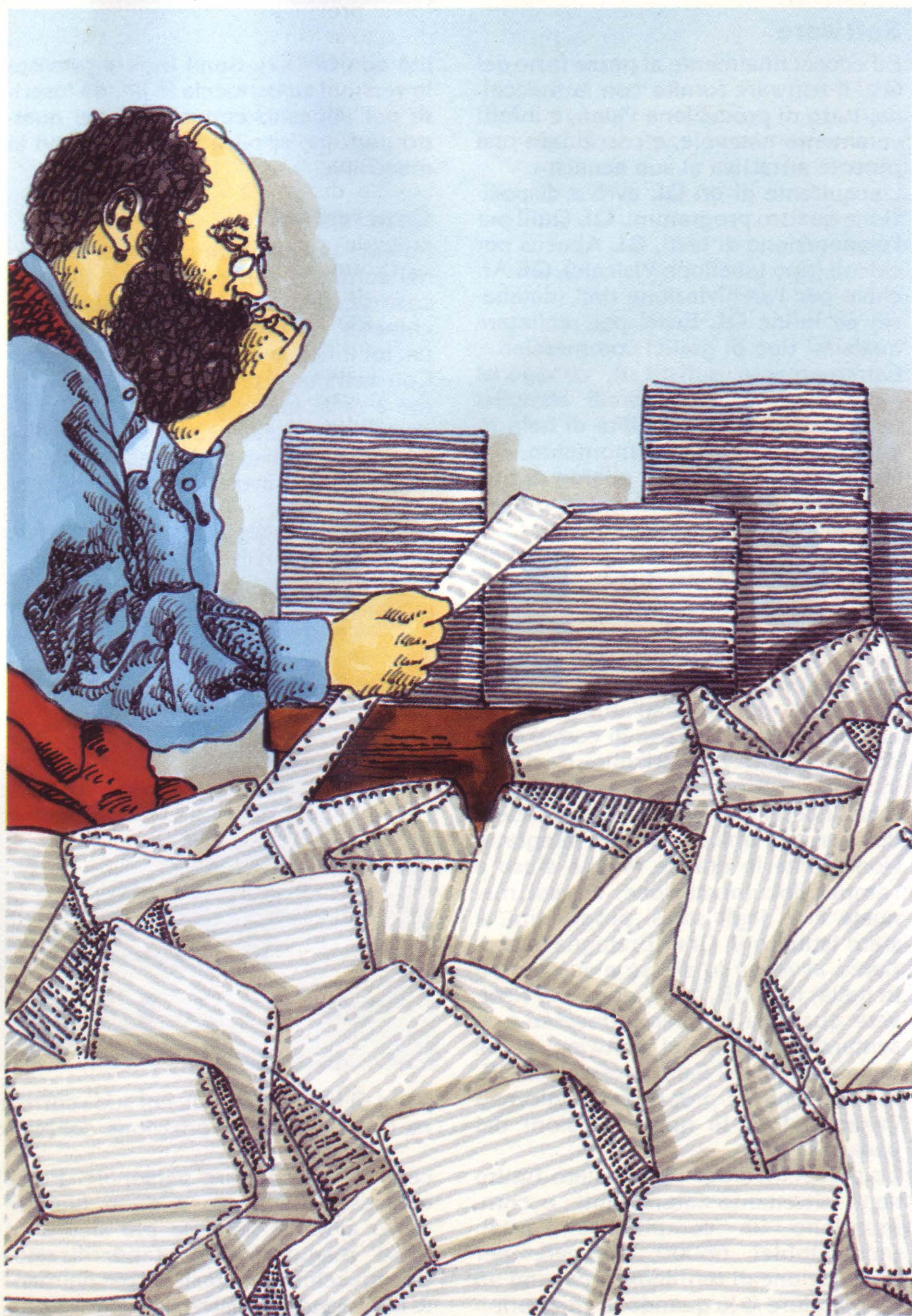


Vista di insieme del QL col suo «corredo». Più in alto, un particolare della macchina che evidenzia la posizione dei Microdrive.

Programmazione strutturata

di Tim Hartnell
trad. di Marcello Spero

Tim Hartnell ci introduce alla programmazione strutturata sullo Spectrum



Molte volte ho scritto articoli o capitoli di libri con l'intento di migliorare la capacità di programmare di coloro che li avessero letti. Ma ogni volta che compilo un elenco delle "cose da fare quando si programma" o cerco di spiegare le medesime cose a qualcuno, mi torna in mente la storiella del contadino. Costui, avvicinato da un giovane che propagandava corsi di agronomia per corrispondenza, si sentì dire: "Già ora non riesco a mettere in pratica tutte le cose che so sull'agricoltura".

Così vanno le cose anche per la programmazione. So perfettamente di non mettere in pratica tutte le cose che so, di violare spesso tutte le regole, buttandomi nella stesura di programmi anche complessi senza tener conto minimamente dei criteri della programmazione strutturata. Il mio suggerimento è perciò di ricordarsi sempre, nella lettura di questo articolo, che io stesso non sono convinto che quanto ho scritto debba essere accettato come la verità assoluta del grande maestro, con minaccia di perdizione per chi non la segue. Il modo migliore per accostarsi a questo articolo, come a qualsiasi altro materiale dello stesso tipo è con un'attenta lettura seguita da riflessione su ciascun suggerimento appreso. Quindi metterete in pratica solo le cose che vi saranno sembrate davvero utili.

Per prima cosa...

L'idea di fondo che vorrei emergesse da questo articolo è che occorre avvicinarsi alla programmazione strutturata con l'idea di "tracciare il contorno" del programma.

L'idea in sé è semplice, ma può esservi molto utile nella stesura di programmi complessi e involuti. È perfettamente

possibile, ovviamente, che voi siate già in grado di creare programmi anche molto complicati senza aver mai usato questo tipo di approccio, ma anche in questo caso vi consiglio di valutare queste idee, che potranno forse rendervi più semplice il lavoro. Il concetto fondamentale della programmazione strutturata sta tutto nell'idea di programmazione "discendente". La parola stessa vi suggerisce di iniziare definendo a grandi linee lo scopo del programma che vi accingete a stendere. Quindi scrivete una serie di osservazioni (ciascuna potrà in seguito diventare una subroutine) che coprano ognuna delle azioni che volete far compiere al computer. Per chiarire meglio quest'ultimo concetto vi darò un esempio concreto. Eccovi i possibili passi per scrivere un programma di "Tris". Prima, però, devo avvertirvi che il metodo di programmazione strutturata produce listati parecchio più lunghi di quanto non lo sarebbero altrimenti. Ad ogni modo questo svantaggio è largamente controbilanciato dalla facilità di correzione e modifica dei programmi, nonché dalla loro enorme facilità di comprensione da parte di altri programmatori, caratteristica che probabilmente non è propria di molti dei vostri programmi.

INIZIALIZZAZIONE DELLA SCACCHIERA
MOSSA DEL COMPUTER
1 CONTROLLA SE LA CASELLA CENTRALE È VUOTA; ALTRIMENTI PROSEGUI
2 CONTROLLA SE ESISTE UNA MOSSA VINCENTE PER IL COMPUTER; SE C'È FALLA
3 CONTROLLA SE L'AVVERSARIO UMANO PUÒ VINCERE CON LA PROSSIMA MOSSA; SE È COSÌ, BLOCCALO;
4 SE FINORA NON È STATA FATTA ALCUNA MOSSA, CONTROLLA SE È POSSIBILE MUOVERE A CASO; SE CIÒ NON È POSSIBILE, DICHIARA PARTITA PARI
STAMPA LA SCACCHIERA
RICEVI LA MOSSA AVVERSARIA
CONTROLLA SE L'AVVERSARIO HA VINTO; SE SI STOP
TORNA ALLA MOSSA DEL COMPUTER

Se analizzate per un attimo questo schema, vi accorgerete che delinea quali sono i passi che il computer percorrerà che decisioni dovrà prendere e i loro possibili risultati.

In questo modo avete già completato il primo passo nella stesura del vostro programma di "tris". Vi sarete accorti, a questo punto, di come non abbia alcuna importanza che voi non sappiate ancora in che modo in pratica questi passi saranno compiuti dal vostro Spectrum: tutto ciò che è necessario in questo momento è che vi rendiate conto che tutti i passi elencati dovranno, in qualche modo, essere compiuti.

Quindi...

Il passo successivo del procedimento consiste nel trasformare il nostro schema in una serie di chiamate a subroutine. In linguaggi più strutturati che non il BASIC dello Spectrum (come per esempio il BBC BASIC o il Pascal, che è appunto servito da modello ai tecnici che hanno scritto il BBC BASIC) è piuttosto facile richiamare un certo numero di subroutine dall'interno di un ciclo senza fine, di struttura del tipo DO/WHILE o REPEAT/UNTIL. Queste costruzioni permettono la ripetizione di una serie di passi del programma fino a che una certa condizione non sia soddisfatta, o lo stato di un determinato puntatore o flag non sia mutato.

Con il BASIC dello Spectrum dobbiamo invece accontentarci del disprezzato GO TO. Molte delle critiche al GO TO, spesso visto come il massimo insulto alla vera arte della programmazione, provengono dalle prime ed in un certo senso rudimentali versioni del BASIC, dove l'unica istruzione che poteva seguire un IF/THEN era GO TO. Questa situazione comportava salti continui, portando ad un programma tutto "spaghettato" ed orrendamente difficile da seguire (il BASIC fornito con la versione standard del TI 99/4A possiede tuttora questa limitazione, come dimostrano i programmi incredibilmente involuti che potete vedere in qualsiasi libro per il TI).

Oggi, d'altra parte, possiamo fare qualsiasi cosa, dopo un IF/THEN (comprese stranezze come un LIST). Ma le vecchie abitudini, si sa, sono dure a morire e la cattiva fama del GO TO continuerà ancora per un po'. La cosa comunque non ci interessa, poiché abbiamo intenzione di produrre programmi chiari e ben strutturati e il GO TO non interferisce in questi scopi.

Torniamo al secondo stadio del nostro procedimento. È possibile trasformare il nostro schema in una serie di chiamate a subroutine inserite in un ciclo senza fine chiuso da un GO TO, come mostrato in figura 1.

A questo punto, come vedete, abbiamo la struttura portante di un programma di "tris", anche se non sappiamo ancora come questo programma funzionerà in pratica. Possiamo ora iniziare la sua stesura vera e propria, pezzo per pezzo.

Questo tipo di approccio possiede altri due importanti vantaggi. Se ci troviamo di fronte qualcosa che non possiamo, a questo stadio, programmare con esattezza (ad esempio il criterio con cui il computer potrà stabilire chi ha vinto), possiamo semplicemente mettere al posto della routine corrispondente una PRINT "CONTROLLO PER IL VINCITORE", che renderà funzionante il programma mentre noi continuiamo a lavorarci sopra. Infatti, ogni volta che dovrebbe effettuare il controllo, il computer stamperà "CONTROLLO PER IL VINCITORE". Il secondo vantaggio lo troviamo alla fine dello stadio più frustrante, e per molti aspetti meno gratificante, della programmazione. Sebbene riesca ad arrivare piuttosto in fretta a far funzionare il mio programma (anche se il mio primo programma di scacchi mi tenne occupato per quasi sei mesi), passare dalla fase di "programma funzionante" a quella di "programma capace di fronteggiare senza problemi qualsi situazione" può portarvi via lo stesso tempo che avete impiegato per arrivare alla prima versione del programma. Vedrete, comunque, che costruisce un programma nel modo descritto in questo articolo può semplificare in modo considerevole la fase di correzione.

Potrebbe capitarvi, ad esempio, che nel vostro programma di "tris" il computer tenda ad ignorare l'angolo in basso a destra della scacchiera anche quando una mossa in questa posizione potrebbe farlo vincere, o potrebbe non bloccare la vittoria avversaria. Dal ciclo delle subroutine che abbiamo creato sappiamo che la mossa del computer viene fatta fra le linee 1000 e 1999. Questa considerazione restringe di molto il vostro campo di ricerca. Se siete stati così abili da costruire ciascuna routine come un sistema di altre subroutine, ognuna costruita secondo lo stesso metodo visto per le subroutine principali, sarà ancora più semplice rintracciare l'errore.

Sembra difficile

Esaminiamo adesso questa idea, costruire ciascuna subroutine come un insieme di altre subroutine, un po' più in particolare. Ci occuperemo della subroutine che inizia in linea 1000, quella cioè in cui il computer sceglie quale

```
10   REM TRIS
20   GOSUB 9000:REM INIZIALIZZAZIONE
30   GOSUB 1000:REM MOSSA DEL COMPUTER
40   GOSUB 8000:REM CREA LA SCACCHIERA
50   GOSUB 2000:REM RICEVI LA MOSSA AVVERSA-
     RIA
60   GOSUB 8000:REM CREA LA SCACCHIERA
70   SE (IF) NÉ IL COMPUTER NÉ L'AVVERSARIO UMA-
     NO HANNO VINTO, ALLORA (THEN) TORNA (GO
     TO) ALLA LINEA 30
80   PRINT "BRAVO!" O QUALSIASI ALTRO MESSAG-
     GIO
```

Un esempio di programmazione strutturata.

mossa fare, essendo questa la più importante e difficile da scrivere.
La subroutine potrebbe avere questo svolgimento mostrato in figura 2.
Programmi costruiti usando un ciclo illimitato accoppiato a subroutine strutturate a "subroutine dentro le subroutine" può, come vi ho già detto, rendere i programmi molto più lunghi del normale. Sullo Spectrum, comunque, non ci sono grossi problemi di memoria (a differenza dello ZX 81 senza espansione di memoria, dove contare i Byte era l'attività dominante). La maggiore lunghezza dei listati, quindi, sarà più che ripagata dalla chiarezza guadagnata dai vostri programmi e dal minor tempo che impiegherete per produrre programmi corretti di cui andare fieri.

Infine...

Infine, permettetemi di suggerirvi qualche altro schema, che potrete provare a trasformare in programmi completi. Questo vi aiuterà a farvi un'idea più completa del funzionamento di questo metodo.
Come primo esempio consideriamo il famoso programma "Life" di John Conway, che simula nascite, vita e morte di una colonia di cellule. Lo schema potrebbe essere semplicemente questo:

INIZIALIZZAZIONE
DISEGNO DELLA COLONIA
AGGIORNAMENTO
RITORNO AL DISEGNO

Senz'altro penserete che sto prendendovi in giro, poiché lo schema non sembra fornire molti suggerimenti circa la stesura del programma. Supponendo però di conoscere le regole, e le vedremo fra un momento, sarete in grado di produrre un programma di buon livello partendo da questo semplice schema.
Eccovi le regole, che suppongono le cellule in una griglia e l'evoluzione di ogni cellula legata allo stesso dello otto cellule che possono circondarla. Se la posizione centrale è vuota, e ci sono esattamente tre cellule nelle posizioni intorno, una nuova cellula verrà disegnata in questa posizione alla generazione successiva. Se invece la posizione centrale è occupata, e ci sono quattro cellule nelle posizioni intorno, alla generazione successiva la cellula che la occupa morirà. La stessa sorte le toccherà se tutte le posizioni intorno a lei sono vuote. Infine, quando ci sono solo due cellule nelle posizioni attorno a quella considerata la situazione resta immutata. Vediamo ora lo schema di un gioco tipo "abbatti il muro", che potrete utilizzare per realizzare un vostro programma:

INIZIALIZZAZIONE
STAMPA RACCHETTA, PALLINA E MATTONI
MUOVI LA RACCHETTA
MUOVI LA PALLINA (FALLA RIMBALZARE SE È NECESSARIO)
SE LA PALLINA HA URTATO UN MATTONE, ELIMINALO ED AGGIORNA IL PUNTEGGIO
SE LA PALLINA È AL LIVELLO DELLA RACCHETTA, CONTROLLA SE QUESTA PUÒ COLPIRLA;
SÈ È COSÌ VAI A "STAMPA RACCHETTA, PALLINA E MATTONI";
ALTRIMENTI VAI A "RIDUCI IL NUMERO DI PALLE DISPONIBILI"
SE IL NUMERO DI PALLINE DISPONIBILE È MAGGIORE DI ZERO, VAI A "STAMPA RACCHETTA, PALLINA E MATTONI"

Vi accorgerete che iniziando in questo modo l'intero processo di costruzione di un programma diventa di gran lunga più semplice di quanto non sarebbe stato altrimenti. Vi accorgerete anche che impiegherete il vostro tempo in maniera più produttiva di quanto non avreste fatto gettandovi nella stesura del programma senza alcuno schema preliminare.
Ignorate perciò il giovane venditore, e cercate oggi stesso di programmare al livello cui siete in grado.

```
1000 REM MOSSA DEL COMPUTER
1010 LET MOSSA=0:REM SE DIVENTA 1 VUOL DIRE
     CHE È STATA TROVATA UNA MOSSA VALIDA
1020 GOSUB 1200:REM CONTROLLA SE LA CASELLA
     CENTRALE È VUOTA
1030 IF MOSSA=1 THEN RETURN
1040 GOSUB 1400: REM CONTROLLA SE ESISTE UNA
     POSSIBILE MOSSA VINCENTE
1050 IF MOSSA=1 THEN RETURN
1060 GOSUB 1600:REM CONTROLLA SE OCCORRE
     BLOCCARE UNA POSSIBILE VITTORIA AVVERSA-
     RIA
1070 IF MOSSA=1 THEN RETURN
1080 GOSUB 1800:REM CONTROLLA SE È POSSIBILE
     UNA MOSSA QUALSIASI
1090 IF MOSSA=1 THEN RETURN
1100 REM UN RITORNO CON MOSSA=0 VUOL DIRE
     CHE NON ESISTONO MOSSE POSSIBILI
1110 RETURN
```

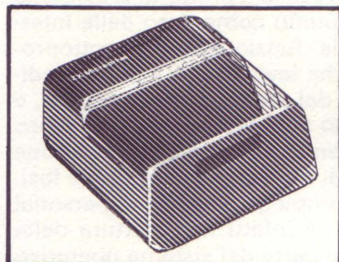
Il contenuto di una subroutine è costituito da tante altre piccole subroutine.



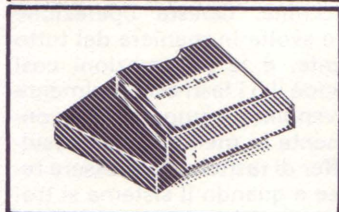
ZX Microdrive



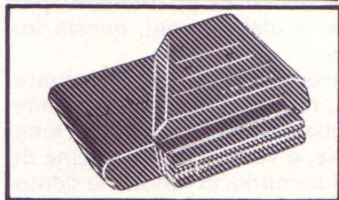
fai crescere il tuo sinclair - Spectrum con le sue eccezionali periferiche!



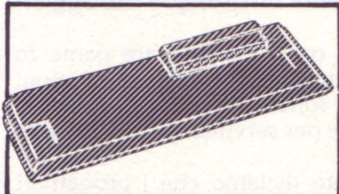
ZX MICRODRIVE
Amplia le possibilità dello ZX Spectrum in quei settori come la didattica e le piccole applicazioni gestionali, dove è necessaria una ricerca veloce delle informazioni.
Ogni cartuccia può contenere:
85 kbyte / 95 kbyte



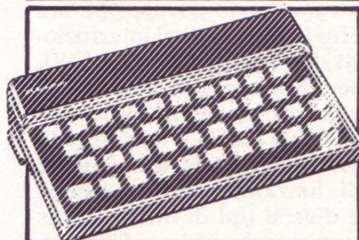
CARTRIDGE
Per ZX Microdrive.
Capacità:
85 kbyte / 95 kbyte
Confezione da 2 pezzi.



ZX INTERFACE 2
Permette di utilizzare le nuovissime ZX ROM, cartucce software e il collegamento per 2 joystick.



ZX INTERFACE 1
Indispensabile per il collegamento dello ZX Microdrive.
Inoltre permette il collegamento fra lo ZX Spectrum e una ampia gamma di periferiche e di altri Sinclair in rete locale.



COMPUTER ZX SPECTRUM
A colori, collegabile ad un televisore a colori o in b/n e ad un normale registratore a cassetta.
32x24 caratteri.
RAM di base: 16 k - 48 k
256x192 punti.
8 colori - 2 luminosità.

sinclair

a casa vostra subito !!

Descrizione	Q.tà	Prezzo unitario	Prezzo Totale
COMPUTER ZX SPECTRUM 16 kbyte		L. 398.000	
COMPUTER ZX SPECTRUM 48 kbyte		L. 499.000	
ZX MICRODRIVE		L. 199.500	
ZX INTERFACE 1		L. 199.500	
ZX INTERFACE 2		L. 95.500	
2 CARTRIDGE		L. 45.000 La coppia	

Desidero ricevere il materiale indicato nella tabella, a mezzo pacco postale contro assegno, al seguente indirizzo:

Nome

Cognome

Via

Città

Data C.A.P.

Desidero ricevere la fattura SI NO

Partita I.V.A. o, per i privati Codice Fiscale

PAGAMENTO:
A) Anticipato, mediante assegno circolare o vaglia postale per l'importo totale dell'ordinazione.
B) Contro assegno, in questo caso, è indispensabile versare l'acconto di Lire 50.000 mediante assegno circolare o vaglia postale. Il saldo sarà regolato contro assegno.
AGGIUNGERE: L. 5.000 per contributo fisso. I prezzi sono comprensivi di I.V.A. 18%

l'unica organizzazione europea per la spedizione di computer e componenti elettronici



EXELCO Via G. Verdi, 23/25
20095 - CUSANO MILANINO - Milano

REM:HW

Hardware

di **Marcello Spero**

Le interruzioni, teoria e pratica sullo ZX Spectrum

Questo articolo si propone di far conoscere l'uso della tecnica di interruzione. Questo potente strumento software, a torto considerato di esclusivo interesse specialistico, può risultare estremamente utile a quanti vogliano ottenere "qualcosa in più" dal proprio elaboratore; infatti, pur richiedendo l'uso del linguaggio macchina per essere utilizzato, può dare grandi vantaggi anche in unione (e all'interno) di programmi BASIC.
Per queste ragioni l'articolo è stato strutturato in modo da essere utilizzato a diversi livelli: nella sua parte generale, e per forza di cose a carattere introduttivo, può interessare chiunque sia curioso o voglia capire di cosa si tratta; i lettori interessati alle applicazioni di questa tecnica per lo ZX Spectrum sono invitati a proseguire la lettura: per poter utilizzare le routine proposte non è necessaria la conoscenza del linguaggio macchina. A quanti invece interessati alle problematiche del I/m dello Z80 sono dedicati alcuni brevi approfondimenti.
Vediamo innanzitutto cosa intendiamo quando parliamo di "interruzione". Tutti noi conosciamo l'uso e lo scopo dei sottoprogrammi, o subroutine che dir si voglia: una procedura (insieme di istruzioni) il cui uso sia necessario più volte nel corso di un programma, viene richiamata mediante un'apposita istruzione, detta appunto "chiamata della subroutine" che compare nei punti del programma ove ne è richiesto l'intervento.
Una volta che questa sia stata eseguita, un'istruzione di "ritorno dalla subroutine", posta in coda alla stessa, farà in modo che l'esecuzione del programma principale riprenda dall'istruzione successiva a quella dichiarata. Questo discorso vale in generale per tutti i linguaggi di programmazione, seppure con le ovvie differenze pratiche. Dover collocare in punti determinati del programma delle istruzioni di chiamata implica però che noi vincoliamo il "momento" della chiamata stessa ai tempi di esecuzione del programma principale. In altre parole, se abbiamo una chiamata a subroutine in linea 500 ed un'altra in linea 1500, non ci importa sapere "quando", rispetto ad un tempo esterno, avverranno queste due chiamate, nè quanto tempo passerà fra la prima e la seconda: ci basta sapere che avverranno esattamente in quei due punti del programma principale.
E se la subroutine contenesse delle istruzioni che richiedono dati ad un congegno esterno che li rende disponibili solo in determinati "momenti", non dipendenti dai tempi di esecuzione del nostro elaboratore?
O se, viceversa, fosse il nostro elaboratore a dover compiere durante la subroutine, operazioni che devono essere svolte in momenti di tempo ben precisi, o ad intervalli regolari?
Ecco che non sapremmo più dove mettere le istruzioni di chiamata. In casi come questi sarebbe molto utile che le chiamate fossero determinate non da un'istruzione presente nel programma principale, ma dal ricevimento da parte dell'elaboratore di un segnale esterno, che gli venisse inviato a tempi prestabiliti. Bene, le interruzioni sono esattamente questo: chiamate di una subroutine in risposta ad un apposito segnale esterno.
Facciamo ora un altro passo avanti; se le chiamate ad una determinata subroutine attuate per mezzo di un'interruzione non dipendono da istruzioni all'interno del programma principale, la medesima subroutine potrà essere utilizzata in programmi diversi, che non necessiteranno di alcuna modifica, se non l'aggiunta della subroutine stessa. D'altra parte, se la subroutine, che possiamo ora chiamare sottoprogramma interrompente (Interruption Handling Routine), non modifica i dati in corso di elaborazione da parte del programma principale, questo potrà proseguire come se non fosse successo nulla, per poi sfruttarne, quando ne avrà la necessità, i prodotti. È evidente a questo punto come l'uso delle interruzioni sia funzionale con sottoprogrammi che lavorano in maniera indipendente dal programma principale, e soprattutto non dipendono, per le loro elaborazioni, dall'avvenuta esecuzione da parte di questo di determinate fasi.
Un tipico impiego, a livello di personal computer, è infatti nella lettura della tastiera da parte del sistema operativo della macchina. Questa operazione può essere svolta in maniera del tutto indipendente, e le informazioni così ottenute (cioè il o i tasti eventualmente premuti) vengono immgazzinate convenientemente in memoria (nel cosiddetto "buffer di tastiera") per essere recuperate se e quando il sistema si troverà nella necessità di ottenere, per proseguire le elaborazioni, questa informazione.
Un tale procedimento può sembrare impreciso, ma se teniamo conto che la lettura della tastiera è un'operazione rapidissima, e viene ripetuta decine di volte ogni secondo, ci rendiamo conto che il sistema avrà sempre dati aggiornati.
Passiamo ora ad esaminare come lo ZX Spectrum si serve delle interruzioni per il suo funzionamento, e cosa possiamo fare per servircene a nostro vantaggio.
Innanzitutto diciamo che i progettisti dello Spectrum hanno dovuto operare una scelta fra i vari "modi di interruzione" previsti dal costruttore della CPU, lo Z80A, per utilizzare quella che più si adattava alle loro necessità. Il microprocessore Z80A (come i suoi fratelli Z80 e Z80B, che differiscono solo nella velocità di funzionamento) possiede infatti due diversi tipi di interruzione: l'interruzione mascherabile e l'interruzione non mascherabile. Il primo tipo può essere eseguito o ignorato a se-



conda che il microprocessore abbia eseguito le istruzioni (in linguaggio macchina) EI (che abilita le interruzioni) o DI (disabilita le interruzioni). Il secondo tipo, invece, non può mai essere ignorato.
Del primo tipo di interruzione esistono tre "modi" diversi, in sostanza metodi diversi di far conoscere al microprocessore la posizione in memoria del sottoprogramma da chiamare. Il secondo tipo di interruzione ha invece un solo modo possibile. Chi fosse interessato ad una conoscenza più approfondita dei vari tipi e modi di interruzione della famiglia Z80 può trovare qui a fianco alcune informazioni; per gli altri è sufficiente sapere che nell'uso normale si fa ricorso all'interruzione mascherabile, mentre quella non mascherabile, d'altronde poco versatile, viene riservata per la gestione di eventi "catastrofici", quali cadute di tensione e simili. Lo ZX Spectrum fa uso delle interruzioni per compiere due operazioni: lettura della tastiera (con aggiornamento della variabile di sistema LAST K) e aggiornamento dell'orologio (variabile di sistema FRAMES). Le due operazioni vengono svolte nel corso della stessa interruzione, e quindi nell'ambito di una sola subroutine interrompente. Per questo è stato scelto di far funzionare la CPU nel modo di interruzione l, cioè il più semplice, che consente di chiamare una sola subroutine e non richiede perciò alcun segnale esterno di indirizzamento. L'unico segnale esterno necessario è quello di richiesta dell'interruzione, che viene inviato alla CPU dall'ULA (il grosso integrato che gestisce i rapporti con l'esterno) al ritmo di 50 volte al secondo. L'ULA ricava questo impulso dalla frequenza della rete elettrica. Ora che abbiamo visto qual è l'impiego normale delle interruzioni nello ZX Spectrum, vediamo cosa è possibile fare per ottenere che altre operazioni vengano portate avanti sotto interruzione. Innanzitutto ci poniamo il vincolo di non apportare alcuna modifica né aggiunta all'hardware esistente: operando solo su software potremo sempre staccare la spina e ripristinare le condizioni originali. Detto questo passimo ad esaminare le modifiche possibili per conquistarci uno "spazio" ove inserire una nostra subroutine. Rimanendo nel modo 1 questo non è possibile, perché l'unico posto disponibile è già occupa-

Tipi e modi di interruzione della famiglia Z80
La famiglia Z80 possiede due tipi di interruzione: mascherabili e non mascherabili, a seconda che possano o meno essere ignorate dal microprocessore che le riceve.

Interruzioni mascherabili

Sono disabilitate: all'accensione, quando il modo di interruzione viene modificato con una istruzione IM, dopo ogni interruzione e dopo l'esecuzione dell'istruzione DI (Disable Interruptions).
Perché siano abilitate deve essere eseguita una istruzione EI (Enable Interruptions). Si articolano in tre modi, selezionabili con istruzioni IM.

to dalla subroutine originale, e tra l'altro corrisponde ad un indirizzo della ROM, pertanto di contenuto modificabile. Il modo 0 in teoria ci consente di scegliere fra 8 indirizzi diversi, ma purtroppo nel nostro caso questi corrispondono tutti a locazioni di memoria all'interno della ROM, e quindi già occupate da routine del sistema.
Ci resta il modo 2: può essere la nostra soluzione, poiché consente di chiamare subroutine situate in una posizione qualsiasi in memoria, e quindi anche in RAM. C'è però un problema; in questo modo la CPU si aspetta dall'esterno metà dell'indirizzo di memoria in cui troverà l'indirizzo a cui trovare la subroutine (è un pò come una caccia al tesoro...); l'altra metà (quella superiore) la trova immagazzinata nel registro I (vettore d'interruzione). La soluzione è comunque a portata di mano: se, infatti, non forniamo alcun indirizzo dall'esterno, verrà assunto come tale il valore massimo possibile, cioè FF Hex, pari a 255 decimale. Variando quindi il valore di I, potremo scegliere tra tutte le locazioni di memoria che terminano con FF: in una di queste porremo il valore dell'indirizzo di inizio di una subroutine scritta da noi per compiere le operazioni che ci interessano.
Per la frequenza di interruzione, invece, non abbiamo scelta: infatti la linea di interruzione è collegata con l'ULA che, come abbiamo visto, invia un segnale ogni 20 msec.
Infine, una considerazione importante per conservare il corretto funzionamento del sistema; se ad ogni richiesta di interruzione la CPU eseguirà la nostra subroutine, la tastiera non verrà più letta, né l'orologio aggiornato. Per includere queste operazioni fra quelle eseguite durante l'interruzione, basta che la nostra subroutine non termini con una istruzione di ritorno al programma principale (RET) ma con un salto verso la "vecchia" subroutine interrompente, posta nella ROM, al termine della quale si trova ovviamente una RET che provocherà il ritorno al programma principale.
Con questi accorgimenti, e costruendo la nostra subroutine in modo che non danneggi nulla all'interno del sistema, potremo ottenere che alcune operazioni di nostra scelta siano portate avanti ogni 20 msec. durante tutto il funzionamento della macchina, cioè sia mentre è in corso di elaborazione un programma che in fase di scrittura e correzione (editing) del programma stesso. Ovviamente queste operazioni dovranno essere di breve durata, diciamo meno di 5 msec., per non rallentare apprezzabilmente il funzionamento dell'elaboratore.
Nella seconda parte vedremo una proposta pratica per utilizzare questo metodo: un orologio con ore, minuti e secondi.

(segue)

Modo 1
È il più semplice. Viene selezionato con l'istruzione IM1, seguita da EI per abilitare le interruzioni. In questo modo quando il microprocessore riceve il segnale di interruzione sulla linea apposita viene eseguito un salto a sottoprogramma verso l'indirizzo di memoria 0038 Hex. Qui dovrà essere presente una subroutine adatta, regolarmente conclusa da un'istruzione RET che provocherà il ritorno al programma principale.
Modo 0
Al ricevimento del segnale di interruzione, il microprocessore risponde con un segnale di "interruzione riconosciuta" su di una linea apposita. In risposta a questo suo "OK" si aspetta il codice di una istruzione lunga un byte sul bus dei dati. Normalmente questo sarà uno dei codici delle istruzioni RST n, dove n può essere una delle otto locazioni di memoria 0,8,10 Hex,20 Hex,28 Hex,30 Hex,38 Hex. RST equivale a CALL nn (è cioè un'istruzione di chiamata di una subroutine) ma è lunga un solo byte invece di tre. In particolare RST 38 Hex (codice FF) chiama la locazione 38 Hex, come nel modo 1, ed è quanto succede se non viene inviato alcun codice in risposta al segnale. È il modo standard all'accensione, e pertanto richiede solo una istruzione EI per funzionare. Può comunque essere selezionata in ogni momento con l'istruzione IMO, seguita da EI.
Modo 2
È il più potente e sofisticato e consente di chiamare una subroutine a qualsiasi indirizzo di memoria.
Per selezionarlo occorre far eseguire la sequenza:

```
IM 2
LD A, n
LD I, A
EI
```

carica il valore n in I, vettore dell'interruzione

Al ricevimento del segnale di interruzione il microprocessore, dopo aver inviato un segnale di "OK", leggerà il codice presente sul bus dei dati e lo considererà il byte di ordine minore di un indirizzo di memoria il cui byte di ordine maggiore è il contenuto di I; prenderà il contenuto di questo indirizzo e chiamerà la subroutine presente nella posizione di memoria da questa specifica.
Non inviando alcun indirizzo sul bus dei dati, il microprocessore vi leggerà, come nel caso del modo 1, il valore FF, che combinerà con il contenuto di I.

Interruzioni non mascherabili

Sono comandate da una linea diversa da quelle mascherabili, e sono sempre abilitate. Il ricevimento di un segnale sulla loro linea provoca sempre e solo un salto a subroutine all'indirizzo 0066Hex.
Sono di uso limitato, in genere in risposta a situazioni di emergenza.
Come viene ottenuta l'emissione sonora nello Spectrum

UN ERRORE PRESENTE NELLA ROM DELLO SPECTRUM

Sebbene non venga utilizzata dallo Spectrum nella sua forma standard, esiste in ROM una routine che si occupa di una eventuale interruzione non mascherabile, qualora la linea venga attivata. Il suo contenuto è:

```
0066 RESET    pus af
              pus hl
              ld hl, (NMIADD)
              ld a,h
              or l
              jr nz,NO-RES
              jp (hl)

0070 NO-RES   pop hl
              pop af
              ret
```

e il suo significato dovrebbe essere:
se la variabile di sistema NMIADD (è quella all'indirizzo 23728, cui il manuale non dà nè un nome nè uno scopo) è a zero non fare niente; se ha un valore diverso, salta all'indirizzo rappresentato dal suo valore. Purtoppo un errore di programmazione della ROM ha fatto sì che l'istruzione

jr z,NO-RES

diventasse

jr nz,NO-RES

rendendo inutile tutta la routine. Infatti l'effetto è ora: se NMIADD è zero salta all'indirizzo corrispondente (cioè zero); se ha valori diversi non fare nulla. In questo modo l'effetto è analogo a quanto si potrebbe ottenere attivando la linea di RESET, oltretutto in maniera più diretta. Pazienza...

SOFTWARE!!

DISTRIBUITO DA

REBIT COMPUTER





A&F software

ANIROG



ATARISOFT

Audiogenic



CDS Micro Systems





Datasoft®





JCE







HEWSON CULSANT





HISOFT PASCAL
DEVPAC



INTERCEPTOR
MICRO'S ©

HesWare

LLAMASOFT!!

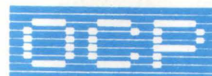
MIKRO-GEN



MELBOURNE



PICTURESQUE





sinclair

PSION



QUICKSILVA

©RABBIT SOFTWARE



SUPERSOFT











UNA VASTA GAMMA DI PROGRAMMI, DI GIOCHI E DI UTILITÁ

DISTRIBUITI DA:



DIVISION OF G.B.C.

Piccola guida all'input dei programmi

I programmi pubblicati da SUPERSINC sono stati accuratamente provati e verificati. In questo modo speriamo di avere ridotto al minimo la possibilità di errori.
Nel caso comunque un programma dovesse presentare malfunzionamenti, pubblicheremo le modifiche da apportare in uno dei numeri immediatamente successivi della rivista.
Per utilizzare i programmi pubblicati è sufficiente digitare i relativi listati sulla tastiera del calcolatore.
Nel caso dello ZX SPECTRUM, abbiamo previsto una semplice decodifica dei tasti da digitare nel modo grafico, per semplificare la comprensione dei listati.
Come è noto, lo SPECTRUM è provvisto di 2 serie di tasti grafici: una prima serie di 16 caratteri grafici predefiniti (i tasti numerici da 1 a 8 e gli stessi "shiftati") e una serie di caratteri definibili dall'utente nell'ambito di un programma (le lettere da A a U).
In entrambi i casi, per ottenere i caratteri desiderati occorre entrare in modo grafico (cursore contrassegnato dalla lettera G lampeggiante) premendo contemporaneamente i tasti CAPS-SHIFT e 9.
Nei nostri listati i caratteri grafici predefiniti sono indicati da una Q seguita dal numero corrispondente al testo che occorre digitare, il tutto racchiuso tra due parentesi graffe.
Ad esempio {G4} significa che occorre digitare il tasto 4, con il cursore in modo grafico.
Analogamente la codifica SG, seguita da un numero da 1 a 8, significa che occorre digitare il relativo tasto numerico premendo contemporaneamente il tasto CAPS-SHIFT.
Ad esempio quando si trova la codifica {SG2}, occorre premere il tasto 2 contemporaneamente al tasto CAPS-SHIFT, ovviamente con il cursore in modo grafico. In entrambi i casi precedenti, quando un simbolo grafico deve essere digitato più volte, i caratteri G o SG della codifica sono preceduti da un numero che specifica quante volte va premuto il tasto grafico indicato.
Così ad esempio {8G5} significa che il tasto grafico 5 va digitato 8 volte e analogamente {4SG1} indica che il tasto grafico 1 e CAPS-SHIFT devono essere battuti 4 volte.
I caratteri grafici definibili (le lettere da A a U in modo grafico) hanno una codifica semplificata: la lettera corrispondente, sottolineata.
Quando in un listato viene presentata, ad esempio, una A sottolineata, occorre entrare in modo grafico (al solito premendo contemporaneamente i tasti CAPS-SHIFT e 9) e quindi digitare semplicemente il tasto che contrassegna la lettera A.

Quando leggete	Premete	Vedrete
{G1}	1	
{G2}	2	
{G3}	3	
{G4}	4	
{G5}	5	
{G6}	6	
{G7}	7	
{G8}	8	
{SG1}	CAPS SHIFT + 1	
{SG2}	CAPS SHIFT + 2	
{SG3}	CAPS SHIFT + 3	
{SG4}	CAPS SHIFT + 4	
{SG5}	CAPS SHIFT + 5	
{SG6}	CAPS SHIFT + 6	
{SG7}	CAPS SHIFT + 7	
{SG8}	CAPS SHIFT + 8	

Se non siete già in modo G, entrateci schiacciando contemporaneamente CAPS SHIFT e 9

Se dovete uscire dal modo G, schiacciate 9

Quando leggete	Premete	Vedrete
A	A	Simbolo grafico definito nel programma in uso.
B	B	
C	C	
D	D	
E	E	
F	F	
G	G	
H	H	
I	I	
J	J	
K	K	
L	L	
M	M	
N	N	
O	O	
P	P	
Q	Q	
R	R	
S	S	
T	T	
U	U	

Se non siete già in modo G, entrateci schiacciando contemporaneamente CAPS SHIFT e 9

Se dovete uscire dal modo G, schiacciate 9

Attacco missilistico

di **Wolfgang Huebl**
trad. e adatt. **Ivan Cerè**

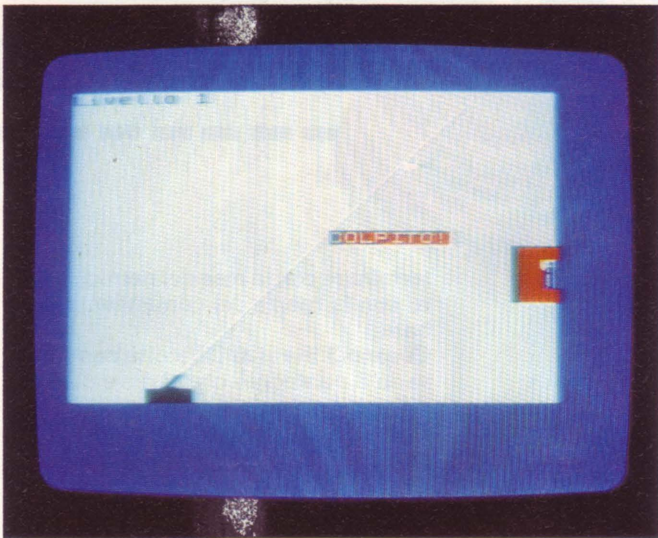
L'obiettivo di questo gioco è di difendere la vostra base da un attacco missilistico alieno.
Quando il programma parte si vedrà il comandante della base protetto dagli scudi sulla parte destra dello schermo. Alla base dello stesso appare un cannone laser, immobile, che spara diagonalmente, con una angolazione fissa. Il vostro compito è distruggere i missili che attraversano lo schermo impedendogli di raggiungere la vostra base.
Ci sono due tipi diversi di missili: quelli neri e quelli gialli. Dovete premere la 'p' per distruggere i missili neri, la 'q' per distruggere quelli gialli.
Potete sparare una sola volta ad ogni missile. Guadagnerete 1 punto per ogni missile nero e 2 punti per ogni missile giallo. Se due missili vengono colpiti contemporaneamente, guadagnerete 6 punti.
Se i missili raggiungono gli scudi ne ditruggeranno una parte. Il gioco può terminare con la resa dei nemici, oppure con la morte del comandante della base.
Ci sono 3 livelli di difficoltà con i quali varia il punteggio che si deve ottenere affiché il nemico si arrenda.
Ecco ora una descrizione del programma:

Linea	Commento
10,25,1000	Contiene la routine per la definizione dei caratteri grafici.
30-80	Contiene una routine per stampare le istruzioni e per scegliere il livello di difficoltà.
105,106,500	Contiene la routine per stabilire dove i missili appariranno.
107-170	Contiene una routine per il movimento dei missili e per permetterci di sparare.
200-250	Contiene una routine che distrugge una parte degli schermi quando un missile lo colpisce.
300-350	Contiene una routine per fare muovere il missile e per controllare se c'è qualcosa sul suo cammino.
500-575	Contiene una routine che permette a due missili di volare assieme e permette a voi di sparare due volte.
700-830	Contiene una routine per il movimento dei due missili e per controllare se c'è qualcosa sul loro cammino.
700	Contiene una routine per quando il comandante viene colpito.



Attacco missilistico
versione per ZX Spectrum 16K/48K

```
10 FOR n=0 TO 39
20 READ d: POKE USR "A"+n,d
25 NEXT n
30 BORDER 5: PAPER 5: CLS
```



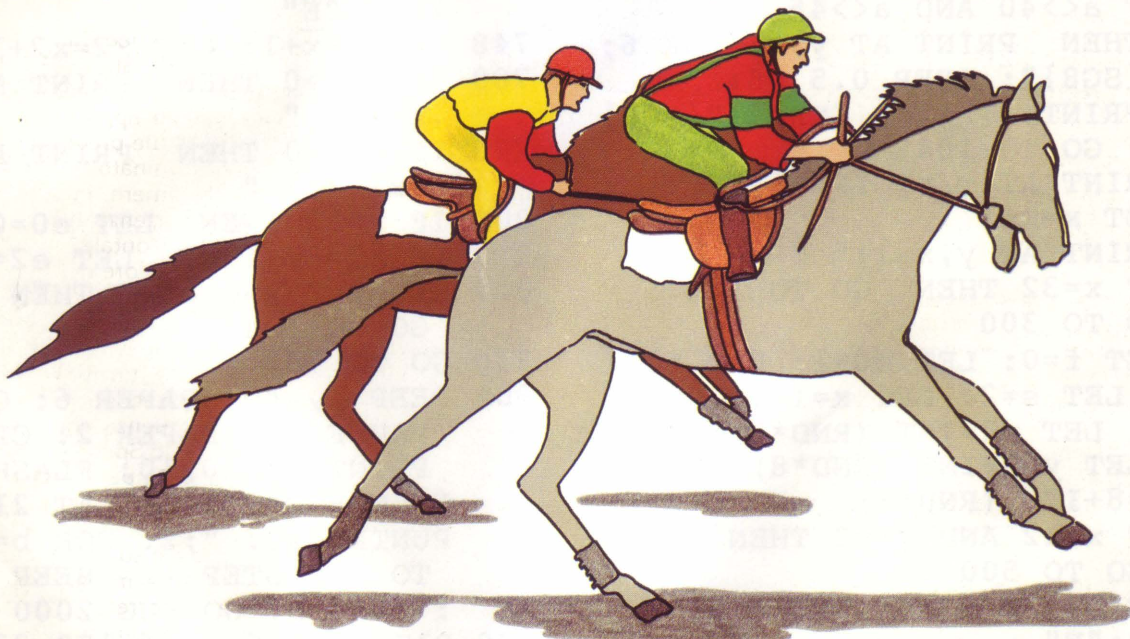

```
40 PRINT AT 0,4; INK 1;"
   {22SG8}"'AT 1,4; PAPER 6;
    INK 1;"{SG8}ATTACCO MISSIL
   ISTICO{SG8}"'AT 2,4; INK 1;
   "{22SG8}"
50 PRINT "Difendi la tua base
   (e la tua  vita) da un att
   acco mmissili-  stico"
52 PRINT : PRINT "Usa le tue a
   rmi a raggio per   distrug
   gere i missili"
54 PRINT : PRINT "Abbatti i mi
   ssili neri con la  'p', e
   guadagnerai un punto"
56 PRINT : PRINT "Usa invece l
   a 'q' per i missili gialli"
   '"Un colpo vale due punti"
58 PRINT : PRINT "Qualche volt
   a affronterai due   missili
    per volta.Colpendoli   en
   trambi guadagnerai 6 punti"
60 PRINT : PRINT "Premi un tas
   to per continuare"
62 IF INKEY$="" THEN  GO TO 62
64 CLS : PRINT "Se un missile
   colpisce la tua   base una
   parte degli scudi pro- tett
   ivi verra' disintegrata"
66 PRINT : PRINT "Se vieni dis
   trutto il gioco   finisce
   ": PAUSE 100
68 CLS : PRINT : PRINT "Ora sc
   egli il livello di    d
   ifficolta'
        (Premi 1,2 o 3)"
70 LET l$=INKEY$
71 IF l$<>"1" AND l$<>"2"
    AND l$<>"3" THEN
    GO TO 70
72 LET l=VAL l$
74 CLS : PRINT "Livello ";l;":
   "'"Devi fare ";10+20*l;" pu
   nti  per fare arrendere il
   nemico"
76 PRINT : PRINT "Hai un solo
   colpo per missile"
78 PRINT : PRINT "Premi 'o' pe
   r iniziare"
80 IF INKEY$<>"o" AND
   INKEY$<>"O" THEN  GO TO 8
   0
101 BORDER 1: PAPER 5: LET f=0:
     CLS
102 LET z=0: LET h=12-l:
     PRINT AT 0,0;"Livello ";l:
     DRAW  INK 5;57,16: PRINT
    AT 20,6;"CD": PRINT AT 21,5
    ;"{3SG8}": PRINT AT h,29;
     INK 2;"{3SG8}": PRINT AT h
    +1,29; INK 2;"{2SG8}";:
     PRINT  INK 1;"A": PRINT
    AT h+2,29; INK 2;"{2SG8}";:
     PRINT  INK 1;"B": PRINT
    AT h+3,29; INK 2;"{3SG8}"
104 IF ATTR (18,8)=47 THEN
     DRAW  INK 5;150,150:
     DRAW  INK 5;-150,-150
105 IF INT (RND*5)>3 THEN
     GO TO 500
106 LET s=1: LET x=INT (RND*z/(
    9-l)): LET y=+INT (RND*8):
     LET f=6*INT (RND+0.25)
107 PRINT AT y,x; INK f;"E"
130 IF s=1 AND INKEY$="p"
     THEN  DRAW 150,150: BEEP 0
    .01,40: DRAW  INK 5;-150,-1
    50: LET s=0
140 IF s=1 AND INKEY$="q"
     THEN  DRAW 150,150: BEEP 0
    .01,30: DRAW  INK 7;-150,-1
    50: LET s=0
144 LET e=ATTR (y,x)
145 IF f=0 AND e=45  THEN
     GO TO 200
147 IF f=6 AND e=47 THEN
     GO TO 200
150 PRINT AT y,x;" ": LET x=x+1
160 IF x>28 THEN  GO TO 300
170 GO TO 107
```

```
200 BORDER 6: PRINT AT 10,17;
     INVERSE 1; FLASH 1; INK 2;
    "COLPITO!": BORDER 1
205 BEEP 1,-20: LET z=z+1+f/6
206 IF z>=10+20*1 THEN
     BORDER 4: PRINT AT 10,7;
     FLASH 1;"IL NEMICO SI ARRE
    NDE": PRINT AT 20,12;"Punte
    ggio finale:";z: FOR b=0
     TO 10: BEEP .3,b: NEXT b:
     GO SUB 2000
217 PRINT AT 10,16;"              ":
     PRINT AT 20,22;""
220 PRINT AT y,x;" "
250 GO TO 104
300 LET a=ATTR (y,x)
308 IF a=41 THEN  PRINT AT y,x;
     INK 0;"{SG8}": GO TO 900
310 IF a<>40 AND a<>45
     THEN  PRINT AT y,x; INK 6;
    "{SG8}": BEEP 0.5,-5:
     PRINT AT y,x; INK 5;"{SG8}
    ": GO TO 104
320 PRINT AT y,x; INK f;"E"
325 LET x=x+1
330 PRINT AT y,x-1;" "
340 IF x=32 THEN  GO TO 104
350 GO TO 300
500 LET f=0: LET e0=1: LET e2=1
    : LET s=2: LET x=INT (RND*5
    ): LET x2=INT (RND*5):
     LET y=8+INT (RND*8): LET y
    2=8+INT (RND*8)
505 IF x=x2 AND y=y2 THEN
     GO TO 500
510 IF e0<>0 THEN  PRINT AT y
    ,x;"E"
520 IF e2<>0 THEN  PRINT AT y
    2,x2;"E"
530 IF s>0 AND INKEY$="p"
     THEN  DRAW 150,150: BEEP 0
    .01,40: DRAW  INK 5;-150,-1
    50: LET s=s-1
540 IF ATTR (y,x)=45  THEN
     LET e0=0
550 IF ATTR (y2,x2)=45 THEN
     LET e2=0
555 PRINT AT y,x;" ": PRINT
    AT y2,x2;" "
560 LET x=x+1: LET x2=x2+1
570 IF e0=0 AND e2=0 THEN
     LET z=z+5: GO TO 200
575 IF x>28 OR x2>28 THEN
     GO TO 700
580 GO TO 510
700 IF (e0=1 AND ATTR (y,x)=41)
     OR (e2=1 AND ATTR (y2,x2)=
    41) THEN  GO TO 900
720 IF e0=1 AND ATTR (y,x)<>4
    0 AND ATTR (y,x)<>45
     THEN  PRINT AT y,x; INK 6;
    "{SG8}": BEEP 0.5,-5:
     PRINT AT y,x;" ": LET e0=0
740 IF e2=1 AND ATTR (y2,x2)
    <>40 AND ATTR (y2,x2)
    <>45 THEN  PRINT AT y2,x2
    ; INK 6;"{SG8}": BEEP 0.5,-
    5: PRINT AT y2,x2;" ":
     LET e2=0
743 IF e0<>0 THEN  PRINT AT y
    ,x;"E"
746 IF e2<>0 THEN  PRINT AT y
    2,x2;"E"
748 LET x=x+1: LET x2=x2+1
780 IF e0<>0 THEN  PRINT AT y
    ,x-1;" "
790 IF e2<>0 THEN  PRINT AT y
    2,x2-1;" "
800 IF x=32 THEN  LET e0=0
810 IF x2=32 THEN  LET e2=0
820 IF e0=0 AND e2=0 THEN
     GO TO 104
830 GO TO 700
900 BEEP 1,-40: PAPER 6: CLS :
     PAUSE 50: PAPER 2: CLS :
     PRINT AT 10,10; FLASH 1;"S
    EI MORTO": PRINT AT 21,20;"
    PUNTEGGIO: ";z: FOR b=0
     TO -40 STEP -2: BEEP 0.2,b
    : NEXT b: GO SUB 2000
1000 DATA 0,6,6,2,14,150,230,254
    ,230,230,230,230,230,230,23
    0,238,0,1,3,7,14,28,56,255,
    224,192,128,0,0,0,0,0,0,192
    ,254,255,254,192,0,0
2000 CLS : PRINT AT 10,2;"Un'alt
    ra partita? (s/n)": PAUSE 0
2207 LET a$=INKEY$
2208 IF a$="" THEN  GO TO 2207
2209 IF a$<>"s" AND a$<>"n"
     THEN  GO TO 2207
2210 IF a$="s" OR a$="S" THEN
     CLS : GO TO 68
2211 IF a$<>"s" OR a$<>"S"
     THEN  CLS : PRINT AT 10,12
    ; BRIGHT 1;"***SALVE***":
     PAUSE 100: NEW
2212 RETURN
```

Ippodromo

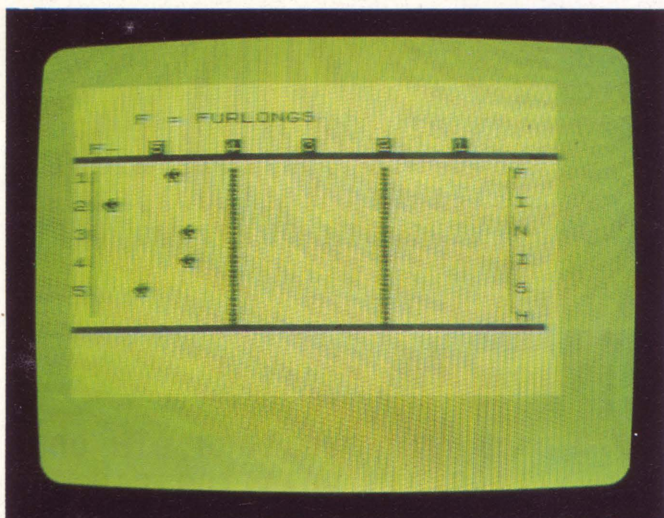
di **N.C. Pearson**
trad. e adatt. **Ivan Cerè**

Questo programma consente una corsa tra 5 cavalli su una distanza di 5 (lunghezze) Furlongs.
All'inizio del programma viene chiesto di scegliere un cavallo o più tra i cinque presenti alla corsa. I cavalli sono presentati con il loro nome, il loro colore e il loro numero; per sceglierli basta scrivere il proprio nome, mentre per passare ad un cavallo successivo bisogna premere la 'x'.
Quando un cavallo vince si è informati se qualcuno ha scommesso su di lui, e si invita lo scommettitore di quel cavallo a ritirare la vincita. Si è anche informati se nessuno ha puntato il cavallo vincente.
Se si desidera cambiare i nomi ai cavalli si può, molto semplicemente, cambiare i nomi alle linee 610-650 inserendo i nuovi nomi che sceglierete per i propri purosangue elettronici.



Ippodromo
versione per ZX Spectrum 16K/48K

```
  2 REM CORSA CAVALLI
  7 REM GRAPHIC H PER I CAVALLI
    ,GRAPHIC F PER LE SIEPI
  8 CLS : BORDER 4: PAPER 4:
     INK 0
 10 GO SUB 467
 20 GO TO 590
 30 BRIGHT 1: CLS : PAPER 4:
     BORDER 4: INK 0
 31 DRAW 255,0: DRAW 0,175:
     DRAW -255,0: DRAW 0,-175
 33 REM ***ISTRUZIONI***
 35 PRINT AT 0,5; FLASH 1;
     INVERSE 1;"CORSA DEI CAVAL
    LI"
 37 FLASH 0
 38 PRINT AT 2,2;"PREMI S P
    ER PARTIRE"
 40 PRINT AT 4,1;" FAI LE TUE S
    COMMESSE ": PRINT AT 6,1;"
    DEVI SCEGLIERE UN CAVALLO
    ": PRINT AT 8,1;"SE NON L'H
    AI ANCORA FATTO ": PRINT
    AT 10,1;" PREMI C PER VEDER
    E I CAVALLI"
 50 LET r=0: LET a=0: LET b=0:
     LET c=0: LET d=0: LET e=0
 52 INPUT m$
 53 IF m$="c" OR m$="C" THEN
     GO TO 583
```




```
57 REM ***DISEGNO***
59 CLS
60 PRINT AT 5,0;"{32G3}"
65 PRINT AT 6,0;"1";AT 6,10;"F";AT 6,20;"F";AT 6,29;"F"
70 PRINT AT 7,10;"F";AT 7,20;"F"
75 PRINT AT 8,0;"2";AT 8,10;"F";AT 8,20;"F";AT 8,29;"I"
80 PRINT AT 9,10;"F";AT 9,20;"F"
90 PRINT AT 10,0;"3";AT 10,10;"F";AT 10,20;"F";AT 10,29;"N"
95 PRINT AT 11,10;"F";AT 11,20;"F"
100 PRINT AT 12,0;"4";AT 12,10;"F";AT 12,20;"F";AT 12,29;"I"
105 PRINT AT 13,10;"F";AT 13,20;"F"
110 PRINT AT 14,0;"5";AT 14,10;"F";AT 14,20;"F";AT 14,29;"S"
115 PRINT AT 15,10;"F";AT 15,20;"F"
120 PRINT AT 16,10;"F";AT 16,20;"F";AT 16,29;"H"
125 PRINT AT 17,0;"{31G3}"
130 PRINT AT 2,4;"L=LUNGHEZZE"
135 PRINT AT 4,1;"L-"
140 PRINT AT 4,5;"5"
145 PRINT AT 4,10;"4"
150 PRINT AT 4,15;"3"
155 PRINT AT 4,20;"2"
160 PRINT AT 4,25;"1"
172 REM ***GIOCO***
180 LET r=INT (RND*6)+0
185 PLOT 10,45: DRAW 0,90
190 PLOT 230,45: DRAW 0,90
200 IF r=1 THEN LET a=a+1: IF r=1 THEN GO TO 250
205 PRINT AT 6,10;"F";AT 6,20;"F"
210 IF r=2 THEN LET b=b+1: IF r=2 THEN GO TO 270
215 PRINT AT 8,10;"F";AT 8,20;"F"
220 IF r=3 THEN LET c=c+1: IF r=3 THEN GO TO 285
225 PRINT AT 10,10;"F";AT 10,20;"F"
230 IF r=4 THEN LET d=d+1: IF r=4 THEN GO TO 300
235 PRINT AT 12,10;"F";AT 12,20;"F"
237 PRINT AT 14,0;"5";AT 14,10;"F";AT 14,20;"F"
240 IF r=5 THEN LET e=e+1: IF r=5 THEN GO TO 320
242 REM ***MOVIMENTO***
250 BEEP 0.03,10: PRINT AT 6,a;"H";AT 6,a-1;" "
255 IF a=29 THEN GO TO 348
265 GO TO 60
270 BEEP 0.03,14: PRINT AT 8,b;"H";AT 8,b-1;" "
275 IF b=29 THEN GO TO 366
280 GO TO 60
285 BEEP 0.03,16: PRINT AT 10,c;"H";AT 10,c-1;" "
290 IF c=29 THEN GO TO 386
295 GO TO 60
300 BEEP 0.03,20: PRINT AT 12,d;"H";AT 12,d-1;" "
310 IF d=29 THEN GO TO 406
315 GO TO 60
320 BEEP 0.03,22: PRINT AT 14,e;"H";AT 14,e-1;" "
325 IF e=29 THEN GO TO 426
340 GO TO 60
346 REM ***ROUTINE VINCITORE***
349 PAUSE 100: CLS
350 IF v$="" OR v$="X" OR v$="x" THEN PRINT AT 10,0;" IL NUMERO 1 HA VINTO, MA NESSUNO HA PUNTATO SU DI LUI": GO TO 460
355 PAUSE 100: CLS
360 PRINT AT 10,0;" IL VINCITORE E' IL NUMERO 1.  "V$;" PUO' RITIRARE LA VINCITA"
```

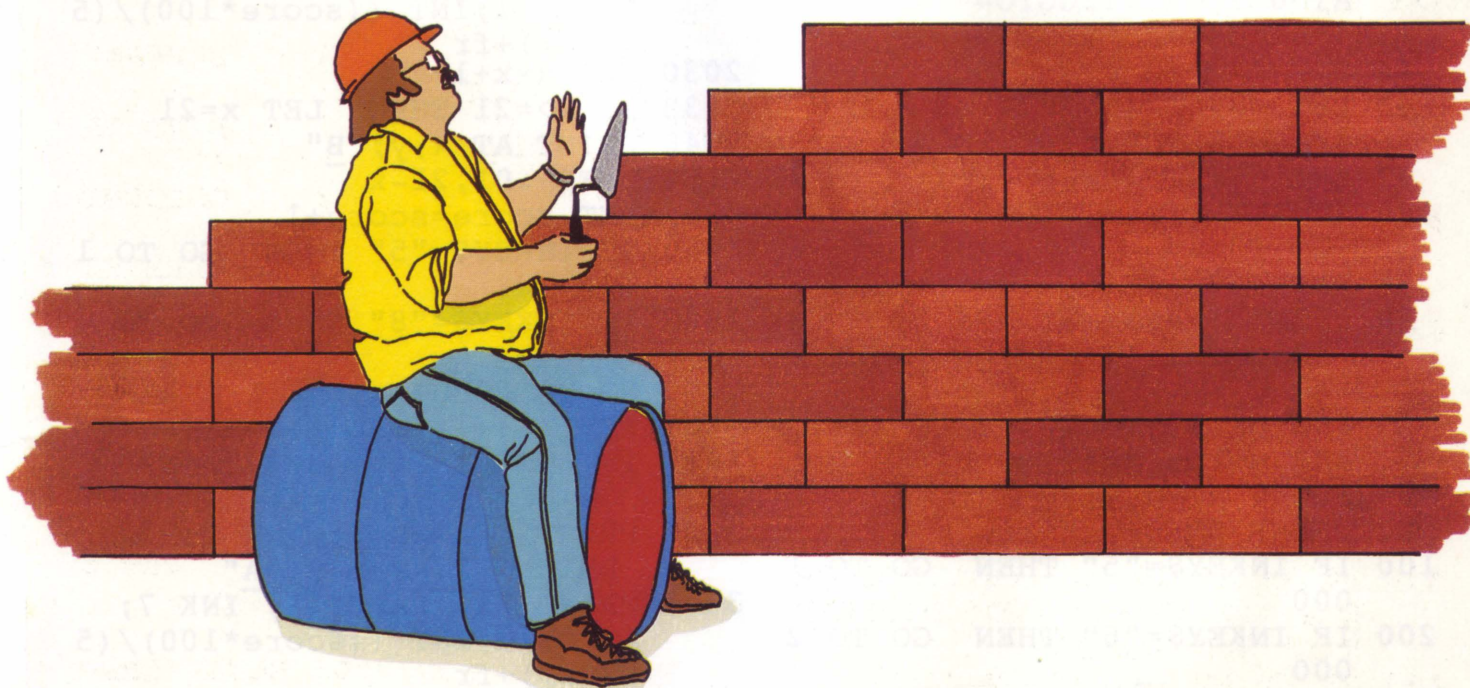
```
365 PAUSE 100: GO TO 460
366 PAUSE 100: CLS
370 IF w$="" OR w$="X" OR w$="X
    " THEN  PRINT AT 10,0;" IL
    NUMERO 2 HA VINTO , MA NES-
     SUNO HA PUNTATO SU DI LUI
    ": GO TO 460
375 PAUSE 100: CLS
380 PRINT AT 10,0;" IL VINCITOR
    E E' IL NUMERO 2 "'w$;" PUO
    ' RITIRARE LA SUA VINCITA"
385 PAUSE 100: GO TO 460
386 PAUSE 100: CLS
390 IF x$="" OR x$="x" OR x$="X
    " THEN  PRINT AT 10,0;" IL
    NUMERO 3 HA VINTO , MA NES-
     SUNO HA PUNTATO SU DI LUI"
    : GO TO 460
395 PAUSE 100: CLS
400 PRINT AT 10,0;" IL VINCITOR
    E E' IL NUMERO 3"'x$;" PUO'
     RITIRARE LA SUA VINCITA"
405 PAUSE 100: GO TO 460
406 PAUSE 100: CLS
410 IF y$="" OR y$="x" OR y$="X
    " THEN  PRINT AT 10,0;" IL
    VINCITORE E' IL 4 MA NES-
     SUNO HA PUNTATO SU DI LUI"
    : GO TO 460
420 PRINT AT 10,0;" IL VINCITOR
    E E' IL 4 "'y$;" PUO' RITIR
    ARE LA SUA VINCITA"
425 PAUSE 100: GO TO 460
426 PAUSE 100: CLS
440 IF z$="" OR z$="x" OR z$="X
    " THEN  PRINT AT 10,0;" IL
    VINCITORE E' IL 5 MA NES-
     SUNO HA SCOMMESSO SU DI LU
    I "
450 PRINT AT 10,0;" IL VINCITOR
    E E' IL 5"'z$;"PUO' RITIRAR
    E LA SUA VINCITA"
460 PRINT AT 14,0;"  VUOI U
    N'ALTRA CORSA  ?": INPUT g$
465 IF g$="S" OR g$="s" THEN
    CLS : GO TO 35
466 NEW
467 REM
468 REM ***CARATTERI GRAFICI***
469 REM
480 FOR j=0 TO 7
490 READ q
500 POKE USR "h"+j,q
510 NEXT j
520 DATA BIN 00001100,BIN 000
    11000,BIN 01111011,BIN 11
    111111,BIN 10111001,BIN 1
    0011100,BIN 00100100,
    BIN 000110110
530 FOR j=0 TO 7
540 READ q
550 POKE USR "f"+j,q
560 NEXT j
570 DATA BIN 00011000,BIN 001
    11100,BIN 00111100,BIN 00
    011000,BIN 00111100,BIN 0
    0111100,BIN 00011000,
    BIN 00111100
580 RETURN
581 REM
582 REM ***SCELTA CAVALLI***
583 REM
590 CLS : PLOT 0,150: DRAW 250,
    0
600 PRINT AT 2,0;" NO   NOME
      COLORE"
610 PRINT AT 4,0;" 1    SUPERMA
    N  GRIGIO"
620 PRINT AT 6,0;" 2    FINE TR
    IM   NERO"
630 PRINT AT 8,0;" 3    GOLDENT
    AILS MARRONE"
640 PRINT AT 10,0;" 4   BREAKE
    R BE GREY/MARRONE"
650 PRINT AT 12,0;" 5    STONEY
     BROAK  MARRONE"
670 PRINT
680 PRINT "PUOI SCEGLIERE QUALS
    IASI CAVALLOSCRIVI IL TUO N
    OME PER PRIMO    PER SCEGLI
    ERE IL CAVALLO 1,    PER S
    ECONDO PER IL CAVALLO 2
    E COSI' VIA.
        SE NON VUOI SCEGLIERE
    IL CAVALLOPREMI X OGNI VOLT
    A"
690 INPUT v$,w$,x$,y$,z$
695 PRINT
700 PRINT "VUOI FARLI PARTIRE ?
    "
750 INPUT G$
800 IF g$="s" OR g$="S" OR g$="
    si" OR g$="SI" THEN
     GO TO 30
810 NEW
```

Bill il muratore

di **C. Elliston**
trad. e adatt. **Ivan Cerè**

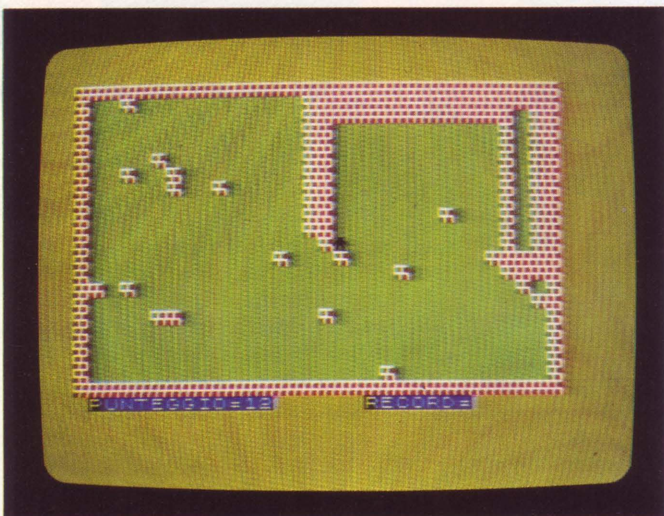
È un gioco semplice ma estremamente difficile da giocare. Si controlla il flusso di mattoni nell'area di gioco usando i quattro tasti cursori. Il movimento dell'omino può essere fermato solamente andando contro un ostacolo o cambiando direzione.
In pratica dovrete cercare di muovere Bill evitando che resti intrappolato. I punti equivalgono alla percentuale di schermo coperta dai mattoni. Una volta raggiunto un punteggio di 80 ci sarà un nuovo schermo, questa volta più difficile per il maggior numero di mattoni già presenti all'inizio del gioco.
Segue una lista delle linee di programma con delle possibili variazioni sul tema:

Linea	Commento
11-14	Definisce i caratteri grafici
32-36	Stampa i muri del bordo
38	Stampa il punteggio
40-50	Stampa il numero di mattoni già presenti all'inizio.
100-5000	Contiene il programma principale di movimento
455	Controlla che l'omino non sia intrappolato
5000-6000	Calcola il punteggio
5025	Converte il numero di mattoni in percentuale di schermo coperta
5053	Questa linea può essere cambiata con un altro numero per cambiare schermo al punteggio voluto
6000-7000	Contengono le istruzioni del gioco



Bill il muratore versione per ZX Spectrum 16K/48K

```
11 FOR n=0 TO 7: READ z:
   POKE USR "a"+n,z: NEXT n
12 DATA 0,119,119,119,0,238,23
   8,238
13 FOR n=0 TO 7: READ z:
   POKE USR "b"+n,z: NEXT n
14 DATA 24,24,60,90,153,36,36,
   102
15 GO SUB 6000
16 LET hi=0: LET fr=0
17 LET h=20
18 LET score=0
20 LET x=10: LET y=15
30 LET a$="A"
32 PRINT BRIGHT 1; PAPER 7;
   INK 2;AT 0,0;"AAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAA"
33 FOR n=1 TO 20
```




```
 34 PRINT  BRIGHT 1; PAPER 7;
     INK 2;AT n,0;"A";AT n,31;"
    A"
 35 NEXT n
 36 PRINT  BRIGHT 1; PAPER 7;
     INK 2;AT 21,0;"AAAAAAAAAAA
    AAAAAAAAAAAAAAAAAAAAA"
 38 PRINT #1; PAPER 1; INK 7;
    AT 0,1;"PUNTEGGIO=";AT 0,19
    ;"RECORD="
 40 FOR b=1 TO h
 45 LET c=INT (RND*20)+1
 46 IF c=10 THEN  GO TO 45
 47 LET d=INT (RND*30)+1
 48 IF ATTR (c,d)>100 THEN
     GO TO 45
 49 BEEP .01,d
 50 PRINT  PAPER 7; INK 2;
     BRIGHT 1;AT c,d;a$: NEXT b

 55 FOR n=-20 TO 20: BEEP .01,
    ABS n+15: NEXT n
 60 PRINT AT x,y;"B"
 80 PAUSE 0
100 IF INKEY$="5" THEN  GO TO 1
    000
200 IF INKEY$="6" THEN  GO TO 2
    000
300 IF INKEY$="7" THEN  GO TO 3
    000
400 IF INKEY$="8" THEN  GO TO 4
    000
450 PRINT AT x,y;"B"
455 IF ATTR (x+1,y)>32 AND
    ATTR (x-1,y)>32 AND ATTR (x
    ,y+1)>32 AND ATTR (x,y-1)>3
    2 THEN  GO TO 5000
500 GO TO 100
1000 REM +++++SINISTRA++++++++
1010 IF ATTR (x,y-1)>32 THEN
     GO TO 100
1020 PRINT  PAPER 7; INK 2;
     BRIGHT 1;AT x,y;"A"
1025 PRINT #1; PAPER 1; INK 7;
    AT 0,11;INT ((score*100)/(5
    99-h))+fr
1030 LET y=y-1
1035 IF y<=0 THEN  LET y=0
1040 PRINT AT x,y;"B"
1050 BEEP .01,21-x
1055 LET score=score+1
1060 IF INKEY$="6" THEN  GO TO 2
    000
1070 IF INKEY$="7" THEN  GO TO 3
    000
1090 GO TO 1000
2000 REM ++++++++GIU'++++++++
2010 IF ATTR (x+1,y)>32 THEN
     GO TO 100
2020 PRINT  PAPER 7; INK 2;
     BRIGHT 1;AT x,y;"A"
2025 PRINT #1; PAPER 1; INK 7;
    AT 0,11;INT ((score*100)/(5
    99-h))+fr
2030 LET x=x+1
2035 IF x>=21 THEN  LET x=21
2040 PRINT AT x,y;"B"
2050 BEEP .01,21-x
2055 LET score=score+1
2060 IF INKEY$="5" THEN  GO TO 1
    000
2080 IF INKEY$="8" THEN  GO TO 4
    000
2090 GO TO 2000
3000 REM ++++++++SU++++++++
3010 IF ATTR (x-1,y)>32 THEN
     GO TO 100
3020 PRINT  PAPER 7; INK 2;
     BRIGHT 1;AT x,y;"A"
3025 PRINT #1; PAPER 1; INK 7;
    AT 0,11;INT ((score*100)/(5
    99-h))+fr
3030 LET x=x-1
3035 IF x<=0 THEN  LET x=0
3040 PRINT AT x,y;"B"
3050 BEEP .01,21-x
3055 LET score=score+1
3060 IF INKEY$="5" THEN  GO TO 1
    000
3080 IF INKEY$="8" THEN  GO TO 4
    000
3090 GO TO 3000
4000 REM ++++++++DESTRA++++++++
```

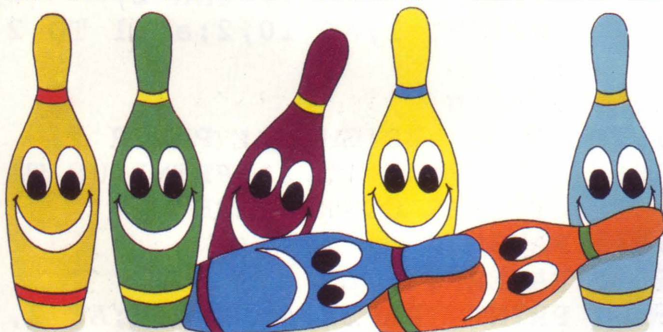
```
4010 IF ATTR (x,y+1)>32 THEN
      GO TO 100
4020 PRINT  PAPER 7; INK 2;
      BRIGHT 1;AT x,y;"A"
4025 PRINT #1; PAPER 1; INK 7;
     AT 0,11;INT ((score*100)/(5
     99-h))+fr
4030 LET y=y+1
4035 IF y>=31 THEN  LET y=31
4040 PRINT AT x,y;"B"
4050 BEEP .01,21-x
4055 LET score=score+1
4070 IF INKEY$="6" THEN  GO TO 2
     000
4080 IF INKEY$="7" THEN  GO TO 3
     000
4090 GO TO 4000
5000 REM ++++++PUNTEGGIO+++++
5025 LET pc=INT ((score*100)/(59
     9-h))
5030 LET fr=fr+pc
5035 IF fr>hi THEN  LET hi=fr
5050 PRINT #1; PAPER 1; INK 7;
     AT 0,1;"PUNTEGGIO=";INT pc;
     " ";AT 0,19;"RECORD=";hi;"
     "
5053 IF pc>=80 THEN  GO TO 5060
5054 LET fr=0: LET pc=0: LET h=2
     0
5055 PRINT  PAPER 6; INK 0;
      FLASH 1;AT 2,1;" PREMI UN
     TASTO PER GIOCARE. "
5059 GO TO 5069
5060 LET pc=0: LET h=h+10: LET f
     r=0
5063 PRINT  FLASH 1; PAPER 6;
      INK 0;AT 2,3;"     PREMI
     UN TASTO      ": PRINT
      FLASH 1; PAPER 6; INK 0;
     AT 3,3;"  PER UN NUOVO ROUN
     D       "
5069 PAUSE 0
5070 FOR n=1 TO 20
5080 PRINT  PAPER 4;AT n,1;"
                               "
     : BEEP .01,n
5090 NEXT n
5093 PRINT #1; PAPER 1; INK 7;
     AT 0,11;fr;"   "
5095 GO TO 18
6000 REM +++ISTRUZIONI+++
6010 BORDER 6: PAPER 4: CLS
6015 DRAW 255,0: DRAW 0,175:
      DRAW -255,0: DRAW 0,-175
6020 FOR n=1 TO 9
6030 PRINT  PAPER 7; INK 2;
      BRIGHT 1;AT n,1;"AAAAAAAAA
     AAAAAAAAAAAAAAAAAAAAA"
6040 NEXT n
6050 PRINT  PAPER 7; INK 2;
      BRIGHT 1;AT 10,1;"A
                          A"
6060 FOR n=11 TO 20
6070 PRINT  PAPER 7; INK 2;
      BRIGHT 1;AT n,1;"AAAAAAAAA
     AAAAAAAAAAAAAAAAAAAAA"
6080 NEXT n
6090 LET a$="
          BILL IL MURATORE
            AIUTA BILL AD AMMUCCH
     IARE I MATTONI MA ATTENZION
     E A NON ESSERE INTRAPPOLATO
                             BAA
     AAAAAAAAAAAAAAAAAAAAAAAAAAA"
6095 FOR n=0 TO 150
6100 LET a$=a$(2 TO )+a$(1)
6110 BEEP .05,CODE a$(28)/4
6120 PRINT  PAPER 7; INK 2;
      BRIGHT 1;AT 10,2;a$(1 TO 2
     8)
6130 NEXT n
6134 PRINT  BRIGHT 1; PAPER 1;
      INK 7;AT 10,2;"PREMI UN TA
     STO PER CONTINUARE"
6135 PAUSE 0
6140 PAPER 4: CLS
6150 PRINT  PAPER 4; INK 0;AT 2,
     2;"USA I CURSORI PER MANOVR
     ARE    BILL PER LA MATTONA
     IA."
6160 PRINT  PAPER 4; INK 0;AT 6,
     2;"ALLA FINE DI OGNI PARTIT
     A TI    SARA' DATA LA PERCE
     NTUALE        APPROSSIMATIVA
      DEL CORTILE      COPERTO D
     AI MATTONI."
6170 PRINT  PAPER 4; INK 0;
      BRIGHT 1;AT 15,2;"PREMI UN
      TASTO PER PARTIRE."
6175 DRAW 255,0: DRAW 0,175:
      DRAW -255,0: DRAW 0,-175
6180 PAUSE 0
6190 FOR n=0 TO 40 STEP 5
6200 BEEP .03,n
6210 NEXT n
6220 CLS
7000 RETURN
```

Bowling

di **Andrew Cook**
trad. e adatt. **Ivan Cerè**

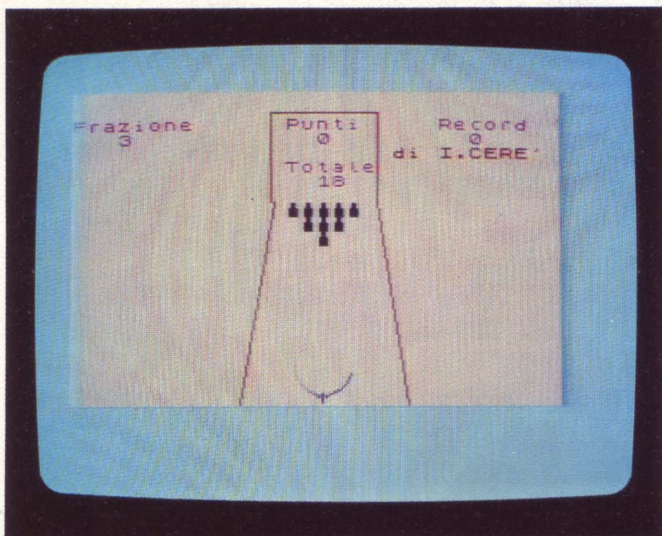
È un programma di uso molto semplice in quanto non ci sono, come succede in molti videogames, 5 o 6 tasti da controllare contemporaneamente. L'unica preoccupazione che avrete è fermare al punto giusto una freccia che scorre sul video.
Quando il programma parte, appare una pedana di bowling.
Una freccia si muove sullo sfondo del video con una velocità che varia con il livello di difficoltà prescelto. Per fermare la freccia basta premere la "Z" e dal punto in cui la freccia è fermata parte una boccia che farà cadere un certo numero di birilli. Ovviamente farete un miglior punteggio con la freccia fermata nel centro della pedana. L'unica abilità richiesta è una buona prontezza di riflessi.
Ecco ora una descrizione del programma:

Linee	Commento
7-13	Seleziona il livello di difficoltà, cioè la velocità di scorrimento della freccia che è determinata dalla lunghezza dei BEEP nella linea 200.
14-70	Stampa la pedana.
540-600	Stampa i birilli alla fine della pedana.
620-800	Contiene la subroutine di disegno della boccia. Questa è ottenuta con il disegno di successivi cerchi il cui centro, a, è determinato dal punto di arresto della freccia.
920-1020	Definisce i caratteri grafici.
2000-2050	Contiene il loop usato per il movimento della freccia.
4000-4055	Queste linee determinano quali e quanti birilli cadono.
5040-5120	Contiene le routine di 'punteggio' e 'record'.
5999-7115	Contiene l'introduzione al programma e le istruzioni del gioco.



Bowling
versione per ZX Spectrum 16K/48K

```
  1 REM +++++BOWLING+++++
  3 BORDER 5: PAPER 7: INK 2:
    LET intro=6000
  4 GO SUB intro
  5 CLEAR : RESTORE
  6 LET hi=0
  7 PRINT AT 10,3;"Scegli livel
    lo di difficolta'";AT 12,5;
    "1 Facile";AT 13,5;"2 Medio
    ";AT 14,5;"3 Difficile"
 10 LET d$=INKEY$: IF CODE d$<4
    9 OR CODE d$>51 THEN
     GO TO 7
 11 IF d$="1" THEN  LET beep=.0
    25
 12 IF d$="2" THEN  LET beep=.0
    125
 13 IF d$="3" THEN  LET beep=.0
    025
 14 CLS : PLOT 88,0: DRAW 18,11
    2
 15 PLOT 176,0: DRAW -16,112
 16 PLOT 104,112: DRAW 0,50
 20 PLOT 160,112: DRAW 0,50:
     DRAW -56,0
 29 LET total=0: LET count=1
 30 LET flag=0: LET score=0:
     LET flaga=0: LET flagb=0:
     LET flagc=0: LET flagd=0
 31 LET a=0: LET chr=920: LET a
    im=2000: LET path=3000:
     LET hit=4000: LET counter=
    5000
 35 PRINT AT 2,14; INK 3;"Punti
    ";AT 3,16;score
 37 PRINT AT 5,14; INK 3;"Total
    e";AT 6,16;total
 38 PRINT AT 2,0; INK 3;"Frazio
    ne"
 40 PRINT AT 3,3; INK 3;count
 50 PRINT AT 2,24; INK 3;"Recor
    d "
 60 PRINT AT 3,26; INK 3;hi
500 RESTORE
520 GO SUB chr
```




```
540 FOR b=14 TO 18
560 PRINT  INK 0;AT 8,b;"A":
    NEXT b
580 FOR b=15 TO 17: PRINT
    INK 0;AT 9,b;"A": NEXT b
600 PRINT  INK 0;AT 10,16;"A"
610 IF flag=2 THEN  GO TO count
   er
615 GO SUB aim
620 CIRCLE  INK 1;a,20,15
640 CIRCLE  OVER 1;a,20,15
660 CIRCLE  INK 1;a,40,12
680 CIRCLE  OVER 1;a,40,12
700 CIRCLE  INK 1;a,60,9
720 CIRCLE  OVER 1;a,60,9
740 CIRCLE  INK 1;a,80,6
760 CIRCLE  OVER 1;a,80,6
780 CIRCLE  INK 1;a,100,3
800 CIRCLE  OVER 1;a,100,3
850 GO SUB hit
855 IF score=9 AND flag=1
    THEN  PRINT  FLASH 1;AT 3,
   16;"9": BEEP .2,5: BEEP 1,3
   : GO TO counter
857 IF a=131 AND flag=2 THEN
    BEEP .5,5: GO TO counter
860 PRINT AT 3,16; INK 3;score
870 PRINT AT 6,16; INK 3;total
880 PAUSE 100
910 IF a=131 THEN  GO TO 30
915 GO TO 610
918 STOP
919 REM *****chr*****
920 FOR s=0 TO 7: READ x:
    POKE USR "a"+s,x: NEXT s:
    REM graphic A
940 FOR s=0 TO 7: READ x:
    POKE USR "b"+s,x: NEXT s:
    REM graphic B
960 FOR s=0 TO 7: READ x:
    POKE USR "c"+s,x: NEXT s:
    REM graphic C
980 DATA 24,24,24,60,60,60,60,6
   0
1000 DATA 0,0,0,0,31,255,255,31
1020 DATA 0,0,0,0,248,255,255,24
   8
1040 RETURN
1060 STOP
1999 REM *****aim*****
2000 FOR x=14 TO 18: PRINT AT 21
   ,x; INK 0;" ↑": BEEP beep,0
   : IF INKEY$="z" THEN
    GO TO 2050
2005 NEXT x
2010 FOR x=18 TO 14 STEP -1:
    PRINT AT 21,x; INK 0;"↑ ":
    IF INKEY$="z" THEN
    GO TO 2050
2015 NEXT x
2020 GO TO aim
2050 FOR p=13 TO 19: IF
   SCREEN$ (21,p)="↑" THEN
    GO SUB path
2055 NEXT p
2060 RETURN
2999 REM *****path*****
3000 IF p=13 THEN  LET a=114
3005 IF p=14 THEN  LET a=114
3010 IF p=15 THEN  LET a=124
3020 IF p=16 THEN  LET a=131
3030 IF p=17 THEN  LET a=139
3040 IF p=18 THEN  LET a=147
3045 IF p=19 THEN  LET a=147
3050 RETURN
3999 REM *****hit*****
4000 IF a=114 THEN  PRINT  INK 0
   ;AT 8,14;"B";AT 8,15;"B":
    IF flaga=0 AND flagb=0
    THEN  LET score=2: LET fla
   ga=1: LET total=total+score
   : GO TO 4055
4010 IF a=124 THEN  PRINT  INK 0
   ;AT 8,14;"B";AT 8,15;"B";
   AT 8,16;"C";AT 9,15;"C":
    IF flagb=0 THEN  LET score
   =4: LET flagb=1: LET total=
   total+score: GO TO 4055
4030 IF a=131 THEN  PRINT  INK 0
   ;AT 8,14;"B";AT 8,15;"B";
   AT 8,16;"C";AT 8,17;"C";
   AT 8,18;"B";AT 9,15;"C";
   AT 9,16;"C";AT 9,17;"C";
```

```
        AT 10,16;"B": LET score=9-s
        core: LET total=total+score
        : GO TO 4055
4040 IF a=139 THEN  PRINT  INK 0
        ;AT 8,15;"B";AT 8,16;"C";
        AT 8,17;"C";AT 8,18;"B";
        AT 9,17;"C": IF flagc=0
         THEN  LET score=4: LET fla
        gc=1: LET total=total+score
        : GO TO 4055
4050 IF a=147 THEN  PRINT  INK 0
        ;AT 8,17;"C";AT 8,18;"B":
         IF flagd=0 AND flagc=0
         THEN  LET score=2: LET fla
        gd=1: LET total=total+score

4055 LET flag=flag+1
4060 RETURN
4999 REM *****counter*****
5000 PAUSE 50
5010 LET count=count+1
5015 PRINT AT 3,3;count
5020 IF count=11 THEN  GO TO 504
        0
5030 GO TO 30
5040 CLS : PRINT AT 10,10;
         FLASH 1; INK 1;"Punti= ";t
        otal
5042 IF total<20 THEN  PRINT
        AT 12,10; INK 1; FLASH 1;"R
        IPROVA"
5044 IF total>19 AND total<40
         THEN  PRINT AT 12,10;
         INK 1; FLASH 1;"VA MEGLIO"
5047 IF total>39 AND total<60
         THEN  PRINT AT 12,10;
         INK 1; FLASH 1;"BUONO"
5050 IF total>59 AND total<75
         THEN  PRINT AT 12,10;
         INK 1; FLASH 1;"MOLTO BENE
        "
5052 IF total>74 AND total<90
         THEN  PRINT AT 12,10;
         INK 1; FLASH 1;"ECCELLENTE
        "
5053 IF total=90 THEN  PRINT
        AT 12,10; INK 1; FLASH 1;"F
        ANTASTICO"
5055 PAUSE 200
5060 CLS : PRINT AT 10,2;"Vuoi c
        ontinuare? (s/n)"
5070 IF INKEY$<>"s" THEN  IF
        INKEY$<>"S" THEN  IF
        INKEY$<>"n" THEN  IF
        INKEY$<>"N" THEN  GO TO 5
        070
5080 PAUSE 0: IF INKEY$="s" OR
        INKEY$="S" THEN  GO TO 5090
5085 CLS : FOR e=1 TO 130:
         PRINT  INK RND*6;" Ciao ";
        : NEXT e
5087 STOP
5090 IF total>hi THEN  LET hi=to
        tal: GO TO 5100
5095 GO TO 10
5100 CLS : PRINT AT 6,0;"Grandio
        so! Hai battuto il"'"record
        .Scrivi il tuo nome"'"(8 le
        ttere max.) e premi enter"
5110 INPUT a$
5120 GO TO 10
5999 REM *****intro*****
6000 CLS : PRINT  INK 2; FLASH 1
        ;AT 10,11;"BOWLING"
7000 LET a=.2: LET b=.4
7010 BEEP a,5: BEEP a,3: BEEP b,
        1
7020 BEEP b,1: BEEP a,1: BEEP a,
        3: BEEP a,5
7030 BEEP a,6: BEEP b,8: BEEP b,
        8: BEEP a,8: BEEP a,5
7040 CLS : PRINT AT 10,3;"VUOI L
        E ISTRUZIONI ?";AT 11,15;"(
        s/n)"
7050 LET a$=INKEY$
7060 IF a$<>"s" THEN  IF a$
        <>"S" THEN  IF a$<>"n"
         THEN  IF a$<>"N" THEN
        GO TO 7050
7070 IF a$="s" OR a$="S" THEN
         GO TO 7090
7080 GO TO 5
7090 CLS : PRINT AT 8,0;"Giochi
        due volte per frazione"'"a
        meno che tu non faccia"'"st
        rike con il primo tiro.":
         PRINT : PRINT : PRINT "Ogn
        i partita dura 10 frazioni.
        ": PRINT : PRINT : PRINT "P
        remi z per fermare la frecc
        ia."
7100 PRINT AT 20,0; FLASH 1;"PRE
        MI UN TASTO"
7110 IF INKEY$="" THEN  GO TO 71
        10
7115 GO TO 5
```

Ok Corral

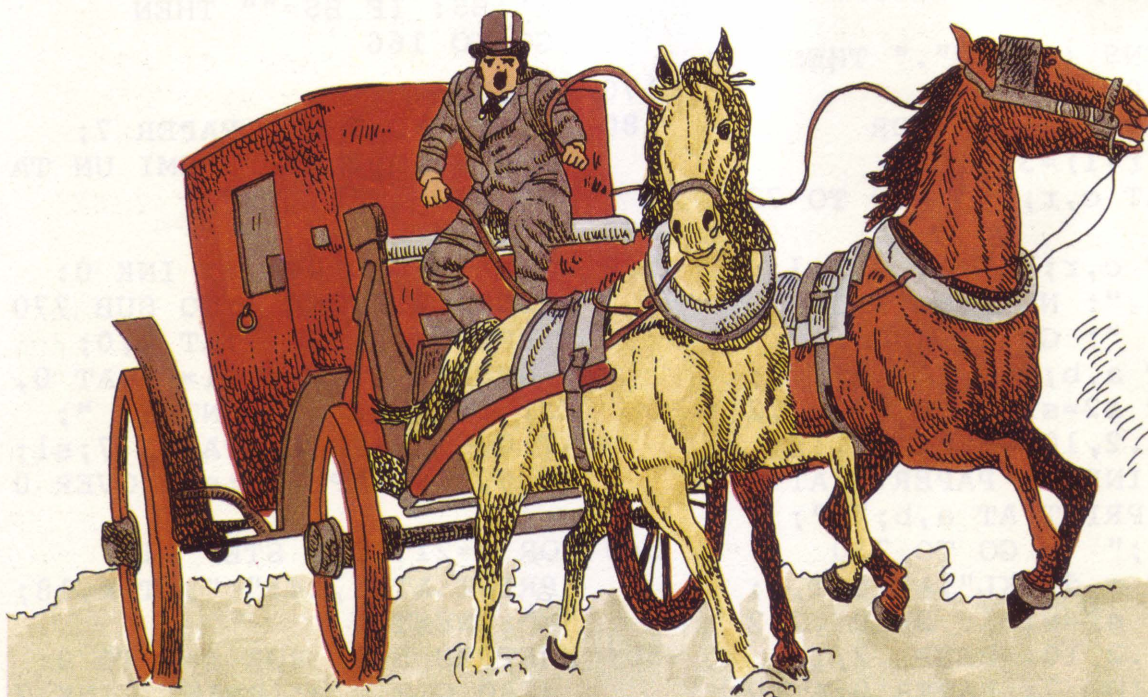
di **R. Page**
trad. e adatt. **Ivan Cerè**

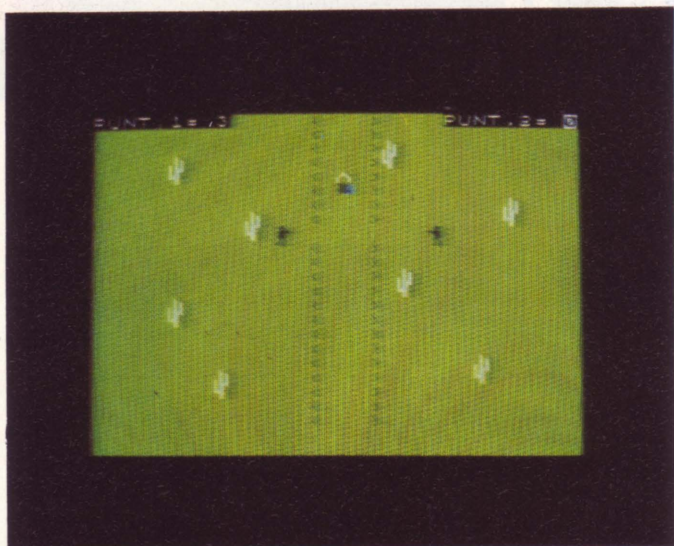
Questo è un gioco per due persone. Essenzialmente è basato su un duello fra i due giocatori che controllano un cowboy ciascuno, piazzati uno sulla destra e l'altro sulla sinistra dello schermo.
Nel mezzo c'è una strada e sparsi per i campi crescono dei cactus, in più sulla strada c'è una diligenza che si muove su e giù. Un cowboy non può essere colpito se si trova protetto da un cactus o dalla diligenza.
Scoprirete anche che i vostri colpi non attraversano completamente lo schermo. Non è quindi una buona tattica starsene nascosti sperando che il vostro avversario si scopra imprudentemente: questo non è un gioco per codardi.

Linee	Commento
10-50	Spara un colpo e controlla se uno dei giocatori è stato colpito
90-200	Stampa le istruzioni del gioco
530-600	Disegna lo schermo
610-760	Contiene il programma principale e controlla se c'è un vincitore
770-830	Definisce i caratteri grafici e assegna i valori alle variabili usate

Per vincere il duello bisogna colpire 5 volte l'avversario.

Eccovi il listato del programma:





Ok corral
versione per ZX Spectrum 16K/48K

```
  3 GO TO 90
  5 REM SPARA E CONTROLLA COLPO
 10 FOR r=(b+2) TO (b+18)
 15 IF SCREEN$ (c,d)="." THEN
     GO TO 50
 16 IF ATTR (a,r)=39 OR ATTR (a
    ,r)=33 THEN  PRINT AT a,r-1
    ;" ": GO TO 650
 17 PRINT AT a,r-1;" ";AT a,r;
     INK 7;".": NEXT r: PRINT
    AT a,r-1;" ": GO TO 650
 20 RETURN
 25 FOR r=(d-1) TO (d-18)
     STEP -1
 30 IF SCREEN$ (a,b)="." THEN
     GO TO 40
 33 IF ATTR (c,r-1)=39 OR
    ATTR (c,r-1)=33 THEN
     PRINT AT c,r;" ": GO TO 73
    0
 35 PRINT AT c,r;" ";AT c,r-1;
     INK 7;".": NEXT r: PRINT
    AT c,r;" ": GO TO 730
 40 PRINT AT a,b;"IJ";AT a+1,b;
    " ": LET s2=s2+1: BEEP .2,2
    0: BEEP .2,10: BEEP .2,15:
     PRINT  INK 7; PAPER 0;AT 0
    ,31;s2: PRINT AT a,b;" ";
    AT a,b+1;" ": GO TO 730
 50 PRINT AT c,d;"KL";AT c+1,d;
    " ": LET s1=s1+1: BEEP .2,2
    0: BEEP .2,10: BEEP .2,15:
     PRINT  INK 7; PAPER 0;AT 0
    ,8;s1: PRINT AT c,d;" ";
    AT c,d+1;" ": GO TO 650
 90 INK 2: PAPER 6: BORDER 6:
     CLS
100 PRINT "DUELLO NEL VECCHIO W
    EST"
110 PRINT AT 2,1;"IL PRIMO PIST
    OLERO CHE COLPISCE IL PROPR
    IO AVVERSARIO ALMENO    CIN
    QUE VOLTE E' IL VINCITORE
      DEL DUELLO"
120 PRINT AT 6,12;"CONTROLLI"
130 PRINT AT 8,4;"PLAYER-1";
    AT 8,20;"PLAYER-2"
135 PRINT AT 9,14;"FUOCO";AT 11
    ,14;"2";AT 11,15;"<>";AT 11
    ,17;"9"
140 PRINT AT 10,8;"SU-Q";AT 12,
    6;"GIU'-1";AT 14,3;"SIN.-CA
    PS";AT 16,4;"DESTRA-A"
150 PRINT AT 10,20;"0-SU";AT 12
    ,20;"P-GIU'";AT 14,20;"ENTE
    R-SIN.";AT 16,20;"SPACE-DES
    TRA"
160 PRINT : PRINT "NON PUOI ESS
    ERE COLPITO SE TI   NASCOND
    I DIETRO UN CACTUS O      DI
    ETRO LA DILIGENZA SULLA STR
    ADA"
165 INPUT "NOME DEL GIOCATORE U
    NO";A$: IF A$="" THEN
     GO TO 165
166 INPUT "NOME DEL GIOCATORE D
    UE";B$: IF B$="" THEN
     GO TO 166
170 PAUSE 50
180 PRINT  FLASH 1; PAPER 7;
     INK 2;AT 21,3;"PREMI UN TA
    STO PER INIZIARE"
200 PAUSE 0
530 PAPER 4: BORDER 4: INK 0:
     CLS : RESTORE : GO SUB 770
    : PRINT  PAPER 7;AT 0,0;
     INVERSE 1;"PUNT.1= ";AT 0,
    23; INVERSE 1;"PUNT.2= ";
    AT 0,8; OVER 1; PAPER 7;s1;
    AT 0,31; PAPER 7;s2: OVER 0
540 FOR f=21 TO 0 STEP -1:
     PRINT AT f,14;"<";AT f,18;
    ">": NEXT f
564 RESTORE 565: FOR f=1 TO 8:
     READ w,x,y,z: PRINT  INK 7
    ;AT w,x;CHR$ 148;AT y,z;
     INK 7;CHR$ 149: NEXT f
```

Ok Corral

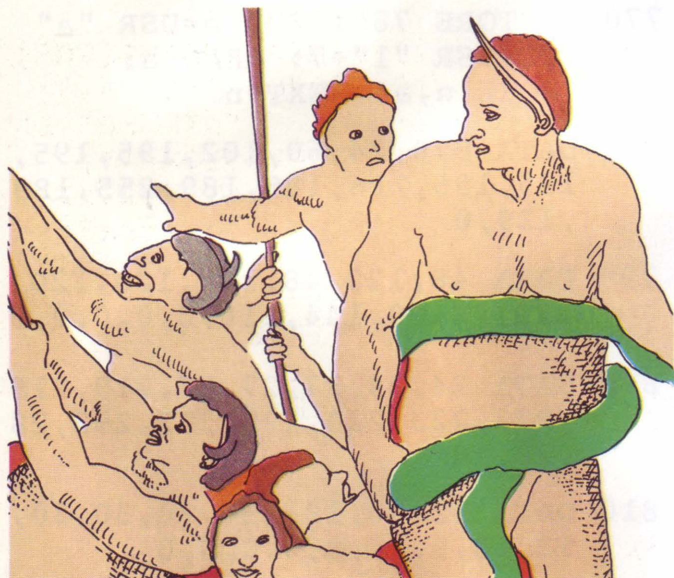
```
565 DATA 11,20,12,20,3,5,4,5,7,
    10,8,10,17,25,18,25,2,19,3,
    19,18,8,19,8,13,5,14,5,6,27
    ,7,27
600 FOR o=19 TO 1 STEP -.5:
     PRINT AT o+1,16;" ";AT o+2
    ,16;" ";AT o-1,16; INK 7;
    CHR$ 144: PRINT AT o,16;
     INK 1;CHR$ 145
610 PRINT  OVER 1;AT a,b;CHR$ 1
    46;AT a+1,b; OVER 1;CHR$ 14
    7
620 LET a=a+(IN 63486=190 AND
    ATTR (a+2,b)=32 AND ATTR (a
    +2,b+1)=32 AND a<19)-(IN 64
    510=190 AND ATTR (a-1,b)=32
     AND ATTR (a-1,b-1)=32
     AND a>2)
625 LET b=b+(IN 65022=190 AND
    ATTR (a,b+2)=32 AND ATTR (a
    +1,b+1)=32 AND b<12)-(IN 65
    278=190 AND ATTR (a,b-1)=32
     AND ATTR (a-1,b-1)=32
     AND b>2)
630 OUT 63486,0: OUT 64510,0:
     OUT 65022,0: OUT 65278,0
645 PRINT AT a,b;CHR$ 146;AT a+
    1,b;CHR$ 147
647 IF INKEY$="2" THEN  GO TO 1
    0
680 PRINT AT c,d; OVER 1;CHR$ 1
    50;AT c+1,d; OVER 1;CHR$ 15
    1
700 LET c=c+(IN 61438=190 AND
    ATTR (c+2,d)=32 AND c<19)-(
    IN 57342=190 AND ATTR (c-1,
    d)=32 AND c>2)
710 LET d=d-(IN 49150=190 AND
    ATTR (c,d-2)=32 AND ATTR (c
    +1,d-1)=32 AND d>20)+(IN 32
    766=190 AND ATTR (c,d+1)=32
     AND ATTR (c-1,d+1) AND d<3
    0)
720 PRINT AT c,d;CHR$ 150;AT c+
    1,d;CHR$ 151
725 IF INKEY$="9" THEN GO TO 2
    5
730 IF s1=5 THEN  PRINT
     INVERSE 1;AT 11,10;
     PAPER 7; FLASH 1;a$;" VINC
    E": PAUSE 200: PRINT AT 13,
    3; PAPER 7; INVERSE 1;"PREM
    I UN TASTO PER GIOCARE":
     PAUSE 0: FLASH 0: GO TO 53
    0
740 IF s2=5 THEN  PRINT AT 11,1
    0; PAPER 7; FLASH 1;b$;" VI
    NCE": PAUSE 200: PRINT AT 1
    3,3; PAPER 7; INVERSE 1;"PR
    EMI UN TASTO PER GIOCARE":
     PAUSE 0: FLASH 0: GO TO 53
    0
750 NEXT o
760 PRINT AT 0,16;" ";AT 1,16;"
     ": GO TO 600
770 RESTORE 780: FOR n=USR "a"
     TO USR "l"+7: READ a:
     POKE n,a: NEXT n
780 DATA 0,0,24,60,102,195,195,
    195,195,126,189,189,255,189
    ,129,0
790 DATA 48,120,48,127,120,120,
    120,48,80,144,216,0,0,0,0,0
800 DATA 24,24,27,27,27,219,219
    ,223,223,220,220,252,252,28
    ,28,28
810 DATA 12,30,12,254,30,30,30,
    12,10,9,27,0,0,0,0,0
815 DATA 16,16,95,255,255,95,0,
    0,0,16,16,240,240,0,0,0
817 DATA 0,8,8,15,15,0,0,0,16,1
    6,250,255,255,250,0,0
820 LET a=19: LET b=1: LET c=19
    : LET d=30: LET s1=0: LET s
    2=0
830 RETURN
```

Nel prossimo numero, anche programmi per lo ZX81.

Inferno

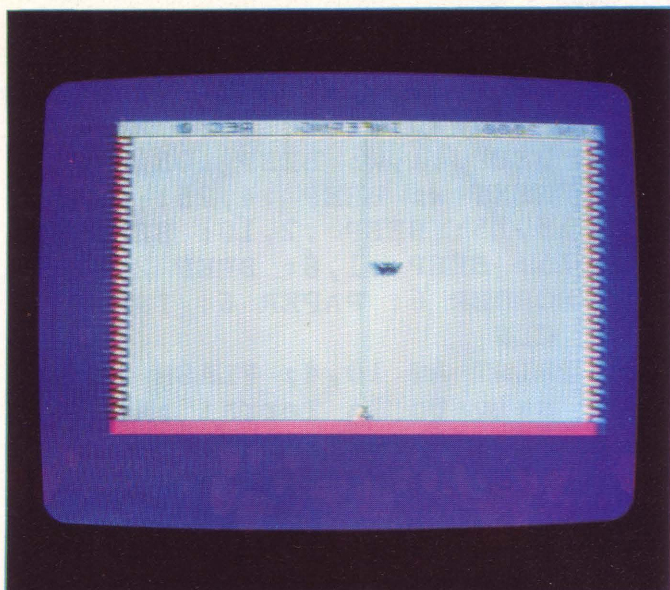
di **Akram Malik**
trad. e adatt. **Ivan Cerè**

In questo programma la vostra missione è difendere la Terra da due ondate di invasori.
La prima ondata non è molto potente e la loro forza distruttiva è abbastanza debole, quindi fino a 5 alieni possono passare attraverso le vostre difese senza provocare la sconfitta totale delle forze terrestri.
Una volta che avete colpito 20 di questi alieni arriverà la seconda ondata. Quest'ultima è molto più potente della prima e quindi solamente 2 di loro possono sfuggire alle vostre difese. Se anche questa volta riuscirete a colpire 20 di questi alieni tutto il genere umano si potrà congratulare con voi per la vostra bravura e il vostro coraggio.
Dovete fare inoltre attenzione a non sprecare i colpi laser. Infatti dopo 100 colpi la vostra energia si esaurirà e non potrete più nulla contro l'invasione.
Naturalmente se, per un motivo o per l'altro, un gran numero di invasori raggiunge la Terra, questa viene distrutta e il gioco termina con un messaggio che informa del triste destino del pianeta.
I tasti di controllo sono: '1'-'5' per muoversi a sinistra, '6'-'0' per muoversi a destra e l'ultima fila di tasti per sparare.



Inferno
versione per ZX Spectrum 16K/48K

```
  1 GO SUB 4000
  2 LET hi=0
  5 LET sc=0
  6 LET an=20: GO SUB 1000
 10 BORDER 1: PAPER 5: INK 2:
    CLS
 20 FOR g=1 TO 21: PRINT AT g,0
    ;"A";AT g,31;"A": NEXT g
 30 FOR h=0 TO 7: PLOT 0,h:
    DRAW  INK 3;255,0: NEXT h
 40 PLOT 0,167: DRAW 255,0
 50 PRINT AT 0,0; INK 3;"PUN ";
    sc;TAB 13; INK 2;"INFERNO";
    TAB 23; INK 0;"REC ";hi
 60 LET lp=100: LET at=5: LET c
    =16: LET a$="BC"
 70 LET i=2: LET q=INT (RND*17)
    +5
 71 IF sc>0 THEN  PRINT AT 0,3;
    " "
 75 IF an=0 THEN  GO TO 1310
 80 PRINT  INK 0;AT i,q;a$
 90 PRINT AT 20,c;" "
 95 IF i=20 THEN  GO TO 270
100 LET c=c+(IN 61438<>255
     AND c<30)-(IN 63486<>255
     AND c>1)
110 PRINT AT 20,c; INK 2;"D"
120 PRINT  OVER 1; INK 0;AT i,q
    ;a$
125 LET i=i+1
130 LET q=q+INT (RND*3)-1+(3
     AND q<-30)-(3 AND q>30)
135 IF q=0 OR q=31 THEN
     GO TO 130
140 PRINT  OVER 1; INK 0;AT i,q
    ;a$: BEEP .01,12
155 IF i=20 AND (q=c OR q+1=c)
     THEN  PRINT AT i,c-1;"EEE"
    : PRINT  INK 2;AT 10,7;
     INK 3;"sei"; INK 0;"stato
    "; INK 2; FLASH 1;"DISTRUT
    TO!": BEEP 2,5: BEEP 1,10:
     PAUSE 150: GO TO 300
160 IF (IN 65278<>255 OR IN 3
    2766<>255) THEN  GO SUB 2
    00: LET lp=lp-1
```



```
165 IF lp=0 THEN  GO TO 6000
170 GO TO 80
200 PLOT  INVERSE 1;c*8+4,17:
    DRAW  INK INT (RND*3);0,14
    8: BEEP .06,16: PLOT
    INVERSE 1;c*8+4,17: DRAW
    OVER 1;0,148
220 IF (i>2 AND i<20) AND (q=c
    OR q+1=c) THEN  LET sc=sc+
    500: PRINT AT 0,4; INK 3;sc
    : BEEP .08,5: PRINT  INK
    INT (RND*3);AT i,c-1;"EEE":
     BEEP .02,15: PRINT AT i,c-
    1;"   ": LET an=an-1:
    GO TO 70
250 RETURN
270 PRINT AT 20,q;"  ": LET sc=
    sc-200: IF sc<0 THEN
    PRINT  INK 3;AT 0,3;sc
275 IF sc>0 THEN  PRINT  INK 3;
    AT 0,0;"PUN ";sc: IF sc>0
    THEN  PRINT AT 0,3;" "
280 LET at=at-1: IF at>0 THEN
    GO TO 70
285 PRINT  INK 2; FLASH 1;AT 10
    ,2;"PAZZO"; INK 1; FLASH 0;
    " ne sono sfuggiti "; INK 3
    ;"ben 5!"
290 BEEP .5,10: BEEP 1,-30:
    BEEP 2,20: BEEP 3,0
310 IF sc>hi THEN  LET hi=sc
320 BORDER 3: PAPER 7: INK 3:
    CLS
330 FOR h=0 TO 40 STEP 8: FOR t
    =40 TO 0 STEP -10: BEEP .07
    ,h: BEEP .04,t: NEXT t:
    NEXT h
```

```
 340 PRINT AT 1,10; FLASH 1;"FIN
     E GIOCO";AT 3,1; FLASH 0;"L
     a terra ora sara' distrutta
     .";AT 10,3;"Il tuo punteggi
     o e':";sc;AT 12,5;"Il recor
     d e':";hi
 341 PRINT AT 16,5;"Un'altra par
     tita? (s/n)"
 342 LET x$=INKEY$
 343 IF x$="" THEN  GO TO 342
 344 IF x$<>"s" AND x$<>"S"
      AND x$<>"n" AND x$<>"N
     " THEN  GO TO 342
 345 IF x$="n" OR x$="N" THEN
      CLS : PRINT AT 10,11;"***B
     YE***": PAUSE 200: NEW
 350 GO TO 5
1000 BORDER 1: PAPER 1: INK 7:
      CLS
1010 PRINT AT 0,10; INK 4;"
     {9SG8}"; OVER 1;AT 0,10;"
              ";AT 1,10;
      PAPER 7;"{SG8}INFERNO{SG8}
     ";AT 2,10;"{9SG8}";AT 4,0;
      PAPER 2; INK 7;"DIFENDI LA
      TERRA DALL'INVASIONE!";
      PAPER 1; INK 7;AT 7,2;"Tas
     ti 1-5 muovi a sinistra";
     AT 9,4;"Tasti 6-0 muovi des
     tra";AT 11,5;"Linea in fond
     o-fuoco";AT 17,8;"Premi un
     tasto"
1100 PAUSE 0: RETURN
1310 FOR y=0 TO 20 STEP 2:
      BEEP .08,y: BEEP .2,5:
      NEXT y
1315 BORDER 2: PAPER 1: INK 7:
      CLS
1317 PRINT AT 11,12;"Fase 2":
      PAUSE 50: CLS
1319 FOR r=1 TO 21: PRINT AT r,0
     ;"F";AT r,31;"F": NEXT r
1320 FOR w=0 TO 7: PLOT 0,w:
      DRAW  INK 4;255,0: NEXT w
1324 PLOT 0,167: DRAW 255,0
1329 PRINT AT 0,0; INK 6;"PUN ";
     sc;TAB 10; INK 5;"INFERNO
     ";TAB 23; INK 4;"REC ";hi
1330 LET at=2: LET c=16: LET al=
     20
1335 LET i=2: LET q=INT (RND*17)
     +5
1340 IF al=0 THEN  GO TO 2000
1380 PRINT  INK 6;AT i,q;"G"
```

```
1390 PRINT AT 20,c;" "
1395 IF i=20 THEN GO TO 1570
1400 LET c=c+(IN 61438<>255
  AND c<30)-(IN 63486<>255
  AND c>1)
1410 PRINT AT 20,c; INK 5;
  BRIGHT 1;"H"
1420 PRINT OVER 1; INK 6;AT i,q
  ;"G"
1425 LET i=i+1
1430 LET q=q+INT (RND*3)-1+(3
  AND q<-30)-(3 AND q>30)
1435 IF q=0 OR q=31 THEN
  GO TO 1430
1440 PRINT OVER 1; INK 6;AT i,q
  ;"G": BEEP .01,15
1450 IF i=20 AND q=c THEN
  PRINT AT i,c; INK (RND*3)+
  4;"E": PRINT INK 7;AT 10,5
  ;"Oh caro!"; INK 7;" Che ";
  INK 7; FLASH 1;"PASTICCIO!
  ": BEEP 3,5: BEEP 3,15:
  PAUSE 50: GO TO 1600
1460 IF (IN 65278<>255 OR IN 3
  2766<>255) THEN GO SUB 1
  500: LET lp=lp-1
1465 IF lp=0 THEN GO TO 6000
1470 GO TO 1380
1500 PLOT INVERSE 1;c*8+4,17:
  DRAW INK (RND*3)+4;0,148:
  BEEP .06,12: PLOT
  INVERSE 1;c*8+4,17: DRAW
  OVER 1;0,148
1520 IF (i>2 AND i<20) AND q=c
  THEN LET sc=sc+500:
  PRINT AT 0,4; INK 6;sc:
  BEEP .04,10: PRINT INK 6;
  AT i,c;"E": BEEP .08,2:
  PRINT AT i,c;" ": LET al=a
  l-1: GO TO 1335
1550 RETURN
1570 PRINT AT 20,q;" ": LET sc=s
  c-200: IF sc<0 THEN
  PRINT AT 0,3; INK 6;sc
1575 IF sc>0 THEN PRINT AT 0,3;
  "": LET at=at-1: IF at>0
  THEN GO TO 1335
1585 PRINT INK 4; FLASH 1;AT 10
  ,4;"FOLLE!"; INK 5; FLASH 0
  ;" Ne sono "; INK 6;"sfuggi
  ti 2!"
1595 BEEP .8,10: BEEP 1,4:
  BEEP 2,-10: BEEP 1,18:
  BEEP 1,-20
1600 GO TO 300
2000 BEEP .08,19: BEEP .2,10:
  BEEP .15,15: BEEP .4,25:
  FOR w=0 TO 20 STEP 2:
  BEEP .1,w: BEEP .05,4:
  NEXT w: BEEP .4,25: BEEP .
  15,15: BEEP .2,10: BEEP .8,
  15: BEEP .1,8: BEEP 2,3
2010 BORDER 6: PAPER 6: INK 0:
  CLS
2020 PRINT AT 10,4; FLASH 1;"Hai
  salvato la Terra!";AT 14,7
  ;"CONGRATULAZIONI!"
2030 PRINT AT 18,2; FLASH 0;"Pre
  mi un tasto per giocare"
2040 PAUSE 0
2050 GO TO 2
4000 FOR n=0 TO 7
4100 READ a: POKE USR "a"+n,a:
  NEXT n
4110 DATA 255,255,192,192,255,25
  5,3,3
4200 FOR n=0 TO 7: READ b:
  POKE USR "b"+n,b: NEXT n
4210 DATA 68,99,50,59,31,31,13,4
4300 FOR n=0 TO 7: READ c:
  POKE USR "c"+n,c: NEXT n
4310 DATA 145,227,166,238,252,12
  0,88,144
4400 FOR n=0 TO 7: READ d:
  POKE USR "d"+n,d: NEXT n
4410 DATA 36,60,24,60,36,102,255
  ,153
4500 FOR n=0 TO 7: READ e:
  POKE USR "e"+n,e: NEXT n
4510 DATA 153,0,36,0,219,0,92,12
  9
4600 FOR n=0 TO 7: READ f:
  POKE USR "f"+n,f: NEXT n
4610 DATA 219,102,219,102,219,10
  2,219,102
4700 FOR n=0 TO 7: READ g:
  POKE USR "g"+n,g: NEXT n
4710 DATA 153,126,90,126,60,36,2
  4,24
4800 FOR n=0 TO 7: READ h:
  POKE USR "h"+n,h: NEXT n
4810 DATA 102,36,36,60,102,231,1
  89,231
5000 RETURN
6000 PRINT INK 2;AT 10,1;"Hai e
  saurito l'energia!": BEEP 5
  ,-30: GO TO 300
```

GP50A E GP50S

le piccole stampanti per tutti i computer

SEIKOSHA





Piccole e compatte dalle prestazioni grandi e generose, le GP50A e GP50S sono realizzate con standard professionali a misura di Personal e Home computer e si impongono quale soluzione ottimale per gli usi hobbystici più disparati a costi incredibilmente sorprendenti. Particolare attenzione merita la GP50S, stampante direttamente interfacciata verso i computer Sinclair ZX81 e Spectrum.

Caratteristiche:

- Stampante ad impatto a matrice di punti da 46 colonne (32 colonne versione GP50S)
- Matrice di stampa 5x8 (7x7 versione GP50S)
- Percorso di stampa monodirezionale (da sinistra a destra)
- Capacità grafiche con indirizzamento del singolo dot
- Possibilità di ripetizione automatica di un carattere grafico
- Velocità 40 caratteri/secondo (35 caratteri/secondo versione GP50S)
- Caratterizzazione: 12 cpi e relativo espanso
- Interfacce: parallela centronics (interfaccia Sinclair versione GP50S)
- Alimentazione carta a frizione (largh. carta fino a 5")
- Stampa originale e 1 copia
- Set di 96 caratteri ASCII
- Consumo 11W (standby) o 17W (stampa)
- Livello di rumore inferiore a 60 dB
- Durata di vita testa: 30 milioni di caratteri
- Peso 1,5 KG
- Dimensioni: 215 (prof.) x 250 (largh.) x 85 (alt.) mm.
- Nastro nero (standard); optionals: rosso, arancio, verde, blu, viola e marrone.

PLOT

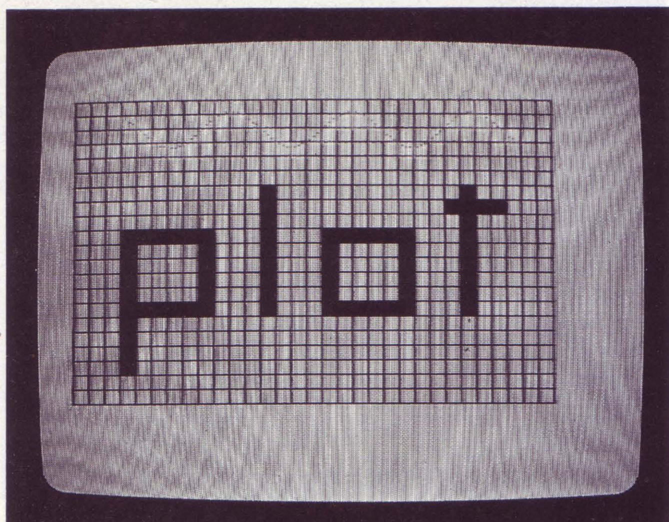
Studio di funzioni sullo ZX Spectrum

di **Carlo Panzalis**

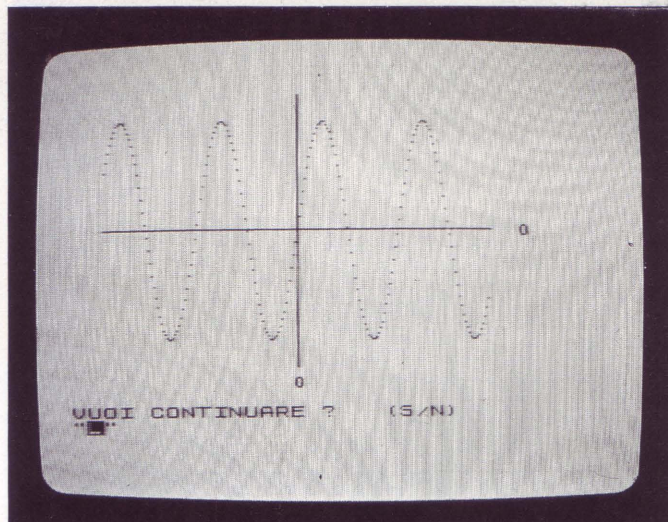
Il programma in queste pagine traccia sul video grafici di funzioni matematiche. Si tratta di un programma semplice, di facile esecuzione, sufficientemente versatile, dato che consente la variazione del range di osservazione dell'andamento della funzione e la sovraimpressione degli assi cartesiani o, a richiesta, di un reticolo ad incremento variabile. "Plot", oltre a consentire la verifica grafica di uno studio di funzione, permette di familiarizzare rapidamente con l'andamento delle più comuni funzioni.
Prima di addentrarci nell'esame delle diverse parti componenti il programma, è opportuno richiamare brevemente, per chi fosse da qualche tempo lontano dai problemi dell'analisi matematica, alcuni fondamenti della stessa.
Le funzioni compatibili con questo programma sono esclusivamente le funzioni monodrome, e non le polidrome o a più valori: per funzione monodroma si intende una qualsiasi corrispondenza che, ad un elemento di un insieme X, associa uno ed un solo elemento di un insieme Y. In concreto, ad ogni valore attribuito alla variabile indipendente x, deve corrispondere un solo valore della variabile dipendente y. Può invece verificarsi il contrario, vale a dire che ad ogni y possono corrispondere più valori di x, senza per questo uscire dall'ambito delle funzioni monodrome. Non possono invece essere inserite nel programma le funzioni che non rispettino tale convenzione, quali generalmente tutte quelle in cui la variabile dipendente y figura con esponente diverso da quello unitario (ad esempio la circonferenza di equazione x↑2 + r↑2, vale a dire y↑2 = r↑2 - x↑2).
Un altro concetto da tenere assolutamente presente è poi quello di campo di esistenza. Tutti ricorderanno che non è possibile l'operazione di divisione se il divisore è zero. Ad esempio, data la funzione y = 1/x in corrispondenza del valore x = 0 si avrebbe y = 1/0, che non ha significato. Nell'introdurre la funzione nel programma, bisognerà fare in modo che i valori che rendono priva di significato la funzione stessa non vengano assunti dalla x del loop che la disegna, onde evitare l'arresto del programma e la comparsa di un errore del tipo "A Invalid Argument".
Quanto detto sin qui per la divisione vale anche per altre operazioni, quali l'estrazione di radice quadrata e comunque di indice pari (radicando < = zero) e il logaritmo (argomento< zero). Nel programma è stato in tal senso predisposto un procedimento che consente di attribuire alla x solo valori coerenti con le premesse sopra esposte, come vedremo in seguito.
Un ultimo consiglio si rende necessario in relazione ad un piccolo limite dello Spectrum: l'operazione di elevamento a potenza non viene eseguita se la base della potenza è un numero negativo, mentre in realtà ci troviamo dinanzi ad una operazione perfettamente lecita. Ricollegandosi alla definizione stessa di potenza, è possibile aggirare l'ostacolo esprimendo ogni volta la potenza come prodotto di basi (x*x invece di x↑2 e così via).
Svolte tali premesse, possiamo ora passare a considerare il listato. In particolare la prima linea del programma consente di assegnare ai 12 indici di una matrice n$ le rispettive 12 stringhe (lunghe al massimo 40 caratteri): le stringhe in questione contengono le coordinate relative in virtù delle quali, muovendosi da uno "starting point", si potranno tracciare i numeri corrispondenti ai diversi step del reticolo che può essere associato al grafico della funzione al fine di renderne più agevole l'esame. Questo aspetto del programma sarà comunque diffusamente chiarito più oltre.
Per quanto concerne la funzione oggetto di studio, essa viene inserita come stringa (f$) in corrispondenza dell'input posto alla linea 60: evidenziamo qui che di essa va sempre inserito il solo secondo membro (Es.: se la funzione data è y = x↑2 - 3*x, si dovrà inserire solo x↑2 - 3*x).
Effettuata tale operazione, attraverso le diverse istruzioni comprese tra le linee 70 e 95 il programma richiede l'eventuale inserimento di condizioni di esistenza. Nel caso in cui la funzione sia definita in tutto il campo reale, nel programma verrà attribuito alla variabile "salto" il valore 1, consentendo così di saltare al momento opportuno un loop designato alla verifica delle condizioni in questione, nidificato in quello che concretamente traccia il grafico.
Nel caso in cui siano invece richieste condizioni di esistenza, esse devono essere inserite come stringhe formulate in modo tale da esprimere l'insieme dei valori per cui la funzione è definita.
Per meglio chiarire gli aspetti di cui abbiamo trattato, consideriamo un esempio: sia data la funzione y = (log x)/(x-1).
I passi da seguire sono:

(i) introdurre la funzione:
(LN x) / (x-1)
(ii) rispondere affermativamente alla richiesta di condizioni.
(iii) inserire "2" per indicare che questo è il numero di condizioni necessarie
(iiii) inserire le due condizioni x<0 e x < > 1, che stanno ad indicare che la funzione è definita solo per valori positivi e maggiori di zero, ad eccezione del valore x = 1 in corrispondenza del quale vi è una discontinuità (peraltro eliminabile).

Chiariti tali punti possiamo proseguire con l'esame del listato: in particolare attraverso i passi compresi fra le linee 100 e 160 si determina il campo di osservazione della funzione, sia in ascissa che in ordinata. Nel caso si intendano mantenere attivi i margini già previsti nel programma (da -12 a +12 in ascissa; da -9 a +9 in ordinata), è sufficiente rispondere affermativamente alla richiesta di input posta alla linea 100. Si rende forse opportuna una breve precisazione in merito: con tali valori si è inteso privilegiare un campo di osservazione abbastanza limitato, poiché è molto spesso in tale ambito che si ritrovano le più rilevanti peculiarità della funzione analizzata. Inotre il rapporto 4/3 che intercorre fra i margini prescelti ripropone il rapporto fra pixel in ascissa e in ordinata: infatti la finestra prescelta per tracciare il grafico presenta 200 pixel in ascissa e 150 in ordinata. È evidente che rispettando quest'ultimo rapporto in sede di definizione dei margini, si evita di alterare il tracciato della funzione. L'utente può comunque, anche in considerazione delle caratteristiche della funzione oggetto di studio, attribuire ai margini qualsiasi valore desiderato, e questo semplicemente rispondendo negativamente alla richiesta di cui sopra, avviando così una procedura di riasse-

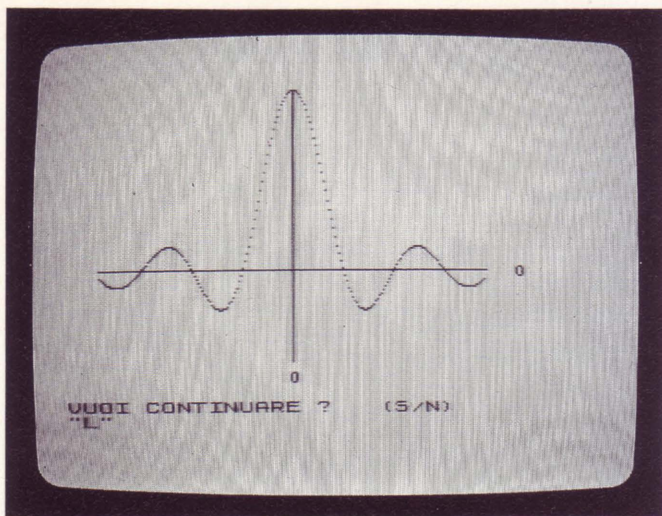


La presentazione di PLOT.

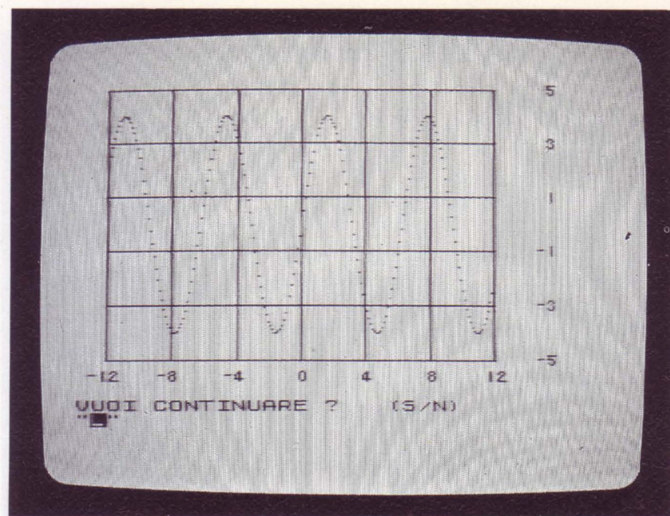


Un esempio di rappresentazione grafica di funzione.

gnazione dei valori ad essi relativi.
Alla linea 180 vengono automaticamente determinati i coefficienti di ingrandimento del grafico, e cioè "c" per la variabile indipendente e "d" per quella dipendente; mentre alla linea 190 viene identificato (indicandolo con "s") lo step del loop che concretamente traccia il grafico. e che si svolge fra le linee 200 e 300. Questi parametri fanno sì che la funzione sia tracciata punto per punto, e ciò indipendentemente dal range di osservazione prescelto.
Ad ogni passaggio alla linea 200 la x (variabile indipendente) assume tutti i valori compresi fra i margini sinistro e destro; è invece il loop che inizia alla linea 220 che, effettuando un salto per tutti gli eventuali valori che rendono priva di significato la funzione sino alla linea 300 (NEXT x), consente di evitare l'arresto del programma, attribuendo così alla x solo valori compresi nel suo insieme di definizione. Le modalità secondo cui tale procedura opera sono le seguenti: di volta in volta vengono considerati sia il valore di x a quel momento, sia le diverse stringhe I $ contenenti le condizioni di esistenza. Se ogni singola proposizione è vera, cioè se il valore considerato di x verifica la condizione, il programma prosegue sino alla linea 250 ove si determina il valore della stringa f$, vale a dire il valore della funzione in corrispondenza di quel dato valore di x. In caso contrario, e cioè se il valore attuale di x non soddisfa la condizione, il programma salta alla linea 300 senza tentare di plottare il punto relativo a tale valore.
Quando il valore della variabile indipendente è compreso nel campo di esistenza della funzione, alle linee 260 e 270 vengono calcolate le coordinate in pixel per l'ascissa e per l'ordinata del singolo punto, mentre le condizioni poste alla linea 280 impediscono l'arresto del programma nel caso in cui tali coordinate siano al di fuori della finestra prescelta per designare il grafico.
Infine la linea 290 imprime il punto secondo quanto determinato nelle fasi precedenti, traslato di 16 pixel in ascissa e di 20 in ordinata al fine di ottenere uno spazio ai lati della finestra, spazio riservato all'inserimento dei numeri relativi ai diversi step del reticolo.
Ultimato il grafico apparirà sulla parte inferiore dello schermo una lista di 3 opzioni (assi, reticolo, fine programma): per passare all'opzione desiderata è sufficiente inserire il numero ad essa corrispondente. Mancando nel Basic Spectrum l'utile ON...GO TO, alla linea 440 il GO TO è stato computato secondo i più semplici fondamenti della logica proposizionale booleana, secondo i quali una proposizione vera ha valore 1, mentre una falsa ha valore 0.
È dunque evidente che solo l'identità verificata fra le tre inserite alla linea 440 assumerà valore 1, mentre le altre due, necessariamente false, assumeranno valore 0, annullando il coefficiente che le precede e consentendo così il salto alla linea desiderata.
Fra le opzioni ora citate, merita ampio chiarimento quella concernente il reticolo: anche l'opzione 1 risulterà chiara dopo tale disamina.
Per tracciare il reticolo il programma viene rinviato alla linea 700: in questa fase viene richiesto un input, corrispondente all'incremento desiderato delle ascisse. In relazione a questo parametro il programma stamperà un dato numero di linee parallele all'asse delle ordinate, numero ottenuto dal rapporto:
(ampiezza intervallo di osservazione) / (incremento ascisse);
questo numero è rappresentato dalla variabile "t" alla linea 710, mentre alla linea successiva "k" determina la lunghezza dell'incremento calcolata in pixel, vale a dire il numero di pixel che separano due assi verticali consecutivi del reticolo. Le condizioni poste alla linea 730, infine, sono atte ad impedire che l'incremento delle ascisse non sia un divisore perfetto dell'ampiezza del range di osservazione.
Procedimento analogo a quello descritto si svolge alle linee 800/830 per tracciare le parallele all'asse delle ascisse: in tal caso "p" svolge la medesima funzione della variabile "t" vista sopra, mentre "f" rappresenta l'incremento in pixel.
Nel gruppo di linee fra la 900 e la 970 si trovano poi le istruzioni per tracciare materialmente il reticolo, le cui caratteristiche sono appunto quelle determinate nelle due precedenti routines.
Una volta terminato il reticolo restano solo da imprimere, in corrispondenza dei punti di partenza di ogni linea verticale ed orrizzontale, i numeri che esprimono il valore del punto di intersezione di ciascuna linea del reticolo con gli assi cartesiani, numeri reali che rendono più agevole l'esame dell'andamento della funzione. La soluzione adottata è tale scopo si è resa necessaria in conseguenza del fatto che, pur consentendo lo Spectrum di trattare simultaneamente pagina testo e pagina grafica, l'utilizzazione del comando PRINT AT avrebbe condotto ad un risultato impreciso sotto l'aspetto formale e grafico, ma soprattutto inefficiente per il corretto abbinamento fra singolo asse e numero ad esso relativo.
È infatti noto che sia i caratteri standard sia quelli definibili dall'utente possono essere impressi solo nei 22*32 quadratini in cui risulta diviso lo schermo, ciascuno dei quali è composto da 8*8 pixel.
Poiché il numero di linee, sia verticali che orizzontali, è di volta in volta deciso dall'utente attraverso l'inserimento degli incrementi desiderati, è pressoché impossibile che il numero di pixel che separa ciascun asse da quello successivo sia sempre un multiplo di 8, vale a dire della lunghezza/altezza di ogni carattere. Per rispettare questa ultima condizione, l'opzione reticolo variabile non avrebbe potuto essere realizzata, dovendosi trovare ogni linea di reticolo in posizione fissa rispetto alla finestra, limitando la versatilità di questo strumento. Se infatti si volessero mantenere variabili gli step del reticolo, l'effetto risultante sarebbe stato normalmente impreciso e addirittura inattendibile in caso di reticolo molto



Variando opportunatamente la scala, è possibile ottimizzare lo spazio a disposizione.



Schermata dell'opzione reticolo. Notare la corrispondenza delle linee con i rispettivi valori.

fitto: tale circostanza avrebbe infatti condotto alla sovraimpressione di numeri su altri già stampati, rendendo così vano l'ausilio rappresentato da tale opzione.

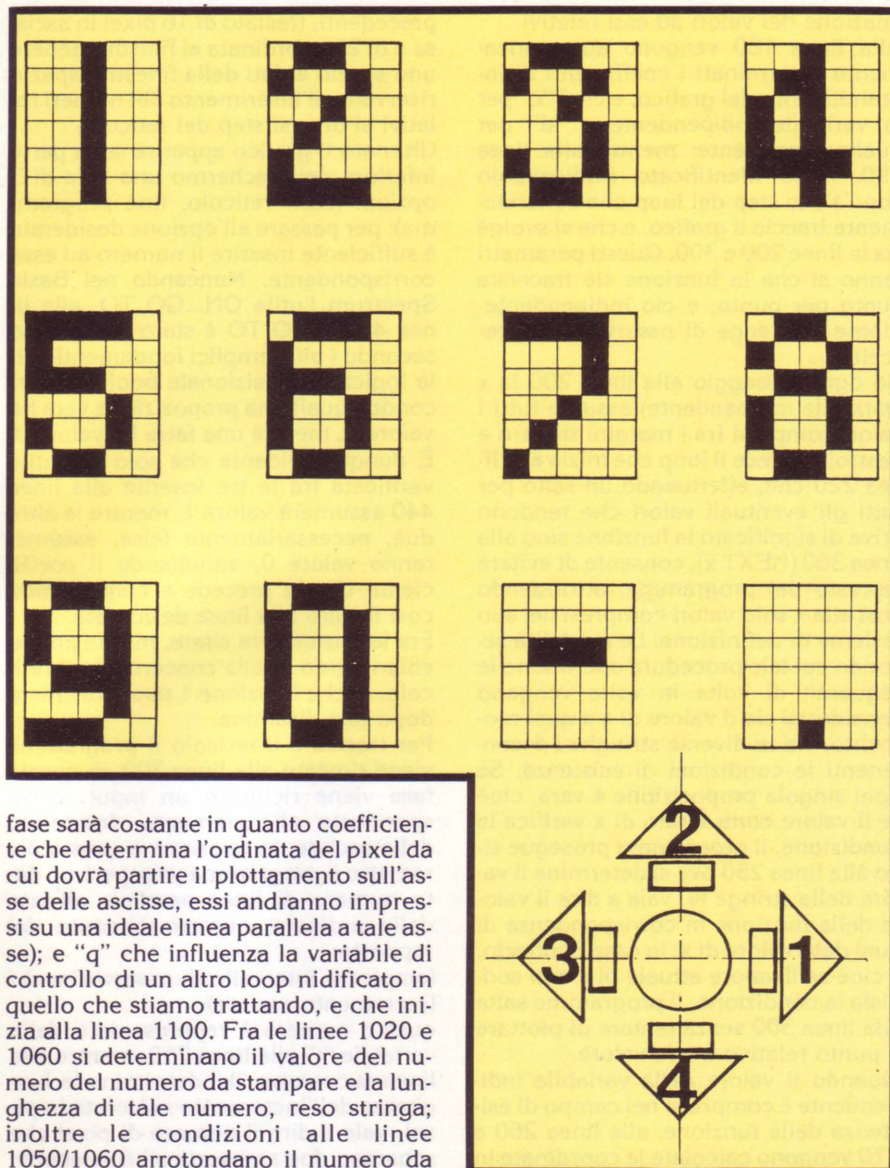
Il problema in questione è stato dunque risolto rinunciando ai caratteri residenti, standard o definibili, componendo 12 simboli (le 10 cifre, il meno, il punto) di dimensioni più piccole (6*5 pixel) che vengono impressi sul video non da PRINT, bensì da PLOT. Entrano cioè in gioco le 12 stringhe cui abbiamo brevemente accennato all'inizio della descrizione del listato, e che contengono le coordinate relative per tracciare i diversi simboli muovendosi da uno starting point determinato dal pixel di partenza di ogni linea, verticale o orizzontale, e dalla lunghezza del numero da rappresentare.

In particolare, per realizzare quanto sopra esposto ci si avvale di 3 routine, vale a dire quella che inizia alla linea 1000 per i numeri riferiti all'incremento sull'asse delle ascisse, quella che inizia alla linea 1200 per i numeri relativi all'incremento delle ordinate, ed infine la routine che inizia alla linea 2000, comune alle due precendenti, e che realizza l'impressione dei numeri richiesti.

Esaminiamo, a scopo semplificativo, lo svolgimento della procedura riferita al solo asse delle ascisse. La variabile "flag", posta uguale a zero alla linea 1005, ha lo scopo di impedire che il primo numero da stampare sullo schermo, nell'angolo in basso a sinistra, esca dallo schermo. Tale variabile verrà continuamente aggiornata nel corso del loop che inizia alla successiva linea di programma, onde evitare anche la sovraimpressione di un numero su altri precedentemente impressi.

Il loop ora citato sfrutta una variabile di controllo ("j") che varia tra zero e il numero e il numero di linee verticali costituenti il reticolo, numero rappresentato da "t". Alla linea seguente sono inizializzate a zero due variabili: "sz", che per tutta la durata di questa fase sarà costante in quanto coefficiente che determina l'ordinata del pixel da cui dovrà partire il plottamento sull'asse delle ascisse, essi andranno impressi su una ideale linea parallela a tale asse); e "q" che influenza la variabile di controllo di un altro loop nidificato in quello che stiamo trattando, e che inizia alla linea 1100. Fra le linee 1020 e 1060 si determinano il valore del numero del numero da stampare e la lunghezza di tale numero, reso stringa; inoltre le condizioni alle linee 1050/1060 arrotondano il numero da rappresentare a zero per valori molto prossimi a tale valore.

Alla linea 1070 la variabile "sv" rappresenta finalmente lo starting point, cioè il punto da cui ciascun numero dovrà essere plottato: questo parametro si ottiene moltiplicando l'incremento in pixel che separa due linee verticali consecutive ("k") e il numero ordinale di ciascuna linea (primo, secondo e così via), sottraendo poi a tale prodotto la metà della lunghezza del numero da stampare reso stringa. I valori costanti in tale algoritmo tengono conto della traslazione della finestra in



ascissa (16 pixel). A questo punto, se lo starting point non è minore di "flag", inizia il loop che attraverso lo slicing e le operazioni di stringa individua di volta in volta il simbolo da rappresentare ed il corrispondente indice relativo alla matrice n$, rinviando il programma alla routine di disegno alla linea 2000. In sostanza a quest'ultima routine si giunge considerando volta per volta una sola cifra (o un simbolo) del numero da rappresentare in corrispondenza di un asse. La linea 2010 contiene due variabili: "as", che rappresenta lo starting point; e "or", che rappresenta l'ordinata di tale punto (costante in questa fase perché "sz" è sempre uguale a zero). Dalla linea successiva si avvia il loop che materialmente traccia il numero: qui "ss" è l'indice della matrice n$, corrispondente al numero trattato. Se, ad esempio, la cifra considerata è 3, il programma considera la stringa n$ (3) che naturalmente contiene le coordinate relative atte a disegnare tale cifra. Attraverso lo slicing vengono dapprima considerati i caratteri che occupano posizioni dispari nella stringa. Quindi, se il primo carattere è uguale a 1, si procede a disegnare il punto di coordinate "as" e "sv"; in caso contrario il punto non viene acceso sullo schermo (se cioè il primo carattere in posizione dispari è zero).
In ogni caso dopo questa operazione viene considerato il primo carattere in posizione pari: se quest'ultimo è 1 viene incrementata di una unità l'ascissa dello starting point, mentre se è 3 tale valore viene ridotto di una unità. Se invece il carattere considerato è 2 o 4 viene allora rispettivamente aumentata o diminuita l'ordinata di quel punto. In sostanza considerando gli elementi che compongono la stringa oggetto di esame, tutti i caratteri in posizione dispari servono a decidere se il punto (as,or) debba o no essere impresso sullo schermo, mentre i caratteri in posizione pari rendono tale punto mobile in quattro direzioni; le differenti combinazioni possibili consentono quindi di disegnare sullo schermo i simboli richiesti che, lo ripetiamo ancora una volta, non sono caratteri standard dello Spectrum. Terminata questa operazione con riguardo alla prima cifra (o simbolo) del numero da rappresentare, il programma torna al loop della linea 1100 ove si individua il successivo simbolo da disegnare, mentre nel contempo vengono aggiornati i valori delle variabili rilevanti ("flag", "sv", "ss") e così via sino al termine del primo numero da stampare, successivamente si identifica il numero da associare al secondo asse verticale (siamo infatti ancora all'interno del loop più esterno, quello che inizia alla linea 1010 e che individua appunto tutti i valori da rappresentare sotto gli assi verticali), e si procede in modo analogo fino all'abbinamento con l'ultima linea verticale.

NOTA

Le dieci cifre, il meno e il punto rappresentano i simboli che vengono impressi in corrispondenza degli assi cartesiani o delle linee orizzontali e verticali che costituiscono il reticolo.
Le coordinate relative per la loro impressione sono contenute nelle 12 stringhe che compongono la matrice n$().
Le quattro frecce stanno invece ad indicare la direzione in cui si sposta il punto mobile che traccia materialmente i simboli, direzione governata dai caratteri in posizione pari nelle stringhe n$(): spostamento verso destra se il carattere 'letto' è 1; spostamento verso l'alto se è 2; e così via.

f$	stringa contenente la funzione di studio
l$()	stringhe contenenti le eventuali condizioni di esistenza
qc	numero di condizioni da porre
ppp	è zero se la condizione l$() non è verificata, altrimenti il suo valore è uno
x, y	rispettivamente variabile indipendente e dipendente
a, b	coordinate in pixel del punto da plottare
ms,md,mb,ma	margini (sinistro, destro, inferiore, superiore) del campo di osservazione
c, d	coefficienti di ingrandimento del grafico
s	step del loop che traccia il grafico
ix, iy	incrementi per le ascisse e per le ordinate, volti a determinare il numero di linee che costituiranno il reticolo che si intende sovraimprimere al grafico della funzione
t	numero di lineee verticali del reticolo
k	distanza in pixel fra due linee verticali del reticolo
p	numero di linee orizzontali del reticolo
f	distanza in pixel fra due linee orizzontali del reticolo
v, z	numeri di volta in volta da rappresentare in corrispondenza, rispettivamente, di una linea verticale o orizzontale del reticolo
v$, z$	numeri v e z resi stringhe
lv, lz	numero di caratteri di v$ e z$
n$()	matrice di 12 stringhe contenenti le coordinate relative per tracciare 12 simboli (10 cifre, il meno, il punto)
sv, sz	rispettivamente ascissa ed ordinata dello starting point, cioè del punto da cui si parte per plottare il numero v o z in corrispondenza di una linea di reticolo
ss	numero indice della stringa n$ da considerare in relazione al simbolo che si deve disegnare
as, or	ascissa ed ordinata del punto che, partendo dallo starting point e muovendosi in una matrice di carattere di 6*5 pixel, traccia il simbolo da rappresentare sullo schermo secondo le istruzioni contenute nelle diverse stringhe n$
flag	variabile che impedisce la sovraimpressione di un numero su altri già tracciati
cont	consente di plottare nuovamente la funzione
salto	consente di saltare il loop di verifica delle diverse condizioni di esistenza nel caso in cui la funzione sia definita in tutto il campo reale

ESEMPI PER L'INSERIMENTO DELLE FUNZIONI E DELLE CONDIZIONI

(1)	Funzione y = SIN x Inserire SIN x Condizioni: nessuna
(2)	Funzione y = SIN x/x Inserire (SIN x/x) Condizioni: una ---- x ><
(3)	Funzione y = log $(x^2 - 3x + 2)$ /x Inserire (LN (x*x - 3*x + 2)) /x Condizioni: una ---- x < > 0 x<1 OR x>2
(4)	Funzione y = xCOS x - SIN x Inserire x*COS x - SIN x Condizioni: nessuna

Ultimate queste operazioni, il programma passa alla nuova routine per tracciare i numeri relativi all'incremento delle ordinate, che parte dalla linea 1200. In tale contesto la procedura seguita è la medesima: cambiano solo poche variabili che debbono tener conto delle differenze. Infatti, mentre nella

routine precendente l'ascissa dello starting point ("sv") dipendeva sia dal pixel di partenza di ciascuna linea verticale sia dal numero di simboli da imprimere, qui il parametro dipende solo dalla lunghezza del numero trattato come stringa. Inoltre, mentre là l'ordinata dello starting point era costante, in questo caso tale variabile (rappresentata da "sz") dipende dal pixel di partenza di ciascuna linea orizzontale. Fatte queste premesse, il procedimento si svolge come già descritto. Termina così l'opzione reticolo; tuttavia il programma non ha ancora termine: infatti rispondendo affermativamente alla richiesta di continuazione che appare al di sotto del reticolo, è possibile sia procedere nuovamente al plottaggio della funzione (modificando eventualmente i margini estremi del campo di osservazione), sia inserire una nuova funzione da analizzare. In conclusione, una volta inserita correttamente una funzione e definito il suo campo di esistenza attraverso le eventuali condizioni inserite come stringhe, è possibile osservarla ripetutamente in diverse "finestre" del campo reale, centrando il range di osservazione sul peculiare andamento di ogni funzione. È cioè possibile, ad esempio, studiare una cubica (y = x↑) dapprima nell'ambito dei margini standard, e poi in margini più ristretti per quanto riguarda l'ascissa (es.: da -3 a +3), mantenendo invece inalterati i margini per l'ordinata (da -9 a +9): l'unica considerazione da tener presente è che in questo modo si altera l'andamento della funzione a causa della diversa scala di misura sui due assi.

Infine, l'ultima parte del programma (da linea 9000) costituisce una pagina grafica di presentazione: salvando il programma con la opzione LINE riferita a quest'ultimo numero di linea, lo si renderà autopartente.

Per concludere intendiamo segnalare che alcune peculiarità di "Plot", quali la composizione di stringhe contenenti informazioni per disegnare simboli grafici di qualsiasi dimensione, o la 'computazione booleana' dei comandi GO TO, possono trovare largo impiego in ogni tipo di programma al fine di superare alcuni piccoli vincoli propri del pur ottimo BASIC Spectrum.

Plot
versione per ZX Spectrum 16K/48K

```
  5 REM ***** PLOT *****
 10 DIM n$(12,40): GO SUB 4000
 15 LET cont=0: LET salto=0
 20 IF cont=0 THEN  GO TO 60
 30 INPUT "VUOI MODIFICARE LA F
    UNZIONE ?";c$
 40 IF c$<>"s" AND c$<>"S"
    AND c$<>"n" AND c$<>"N
    " THEN  BEEP .5,12: GO TO 4
    0
 50 IF c$="n" OR c$="N" THEN
    GO TO 100
 60 CLS : PRINT AT 19,0;"Funzio
    ne": PLOT 0,14: DRAW 63,0:
    INPUT "y=";f$
 65 PRINT AT 21,0;"y=";f$''
 70 INPUT "ci sono condizioni ?
    ";c$
 73 IF c$<>"s" AND c$<>"S"
    AND c$<>"n" AND c$<>"N
    " THEN  BEEP .5,12: GO TO 7
    0
 75 IF c$="n" OR c$="N" THEN
    LET salto=1: GO TO 100
 80 INPUT "quante condizioni ?
    ";qc
 85 DIM l$(qc,20): PRINT '"Cond
    izioni"': PLOT 0,6: DRAW 8
    0,0
 90 FOR u=1 TO qc: INPUT "C. E.
    ";l$(u): PRINT 'l$(u):
    NEXT u
 95 PAUSE 0
 99 REM definizione range varia
    zione
100 CLS : PRINT '"VUOI MARGINI
    STANDARD ?   (S/N)"'"per a
    scissa    ( -12 / 12 )""pe
    r ordinata    ( - 9 /  9 )"
    : INPUT c$
110 IF c$<>"s" AND c$<>"S"
    AND c$<>"n" AND c$<>"N
    " THEN  BEEP .5,12: GO TO 1
    00
120 IF c$="s" OR c$="S" THEN
    LET ms=-12: LET md=12:
    LET mb=-9: LET ma=9:
    GO TO 170
130 PRINT '"ASCISSA :MARGINE SI
    NISTRO ";: INPUT ms:
    PRINT ms
140 PRINT "ASCISSA :MARGINE DE
    STRO  ";: INPUT md: PRINT m
    d
150 PRINT "ORDINATA:MARGINE INF
    ERIORE ";: INPUT mb:
    PRINT mb
160 PRINT "ORDINATA:MARGINE SUP
    ERIORE ";: INPUT ma:
    PRINT ma
170 PAUSE 20: CLS
175 REM definizione coefficient
    i di ingrandimento
180 LET c=200/(md-ms): LET d=15
    0/(ma-mb)
190 LET s=1/c
200 FOR x=ms TO md STEP s:
    REM loop
210 IF salto=1 THEN  GO TO 250
220 FOR u=1 TO qc
225 LET ppp=VAL l$(u)
230 IF ppp=0 THEN  GO TO 300:
    REM eventuali condizioni d
    'esistenza
```

```
240 NEXT u
250 LET y=VAL f$: REM funzione
260 LET a=(x-ms)*c
270 LET b=(y-mb)*d
280 IF b<0 OR b>150 THEN
    GO TO 300
290 PLOT a+16,b+20
300 NEXT x
400 PRINT AT 21,0;"1)ASSI   2)
    RETICOLO    3)USCIRE":
    REM opzioni
410 INPUT r
420 IF r<>1 AND r<>2 AND r
    <>3 THEN  BEEP .5,12:
    GO TO 400
430 FOR h=0 TO 31: PRINT AT 21,
    h;" ": NEXT h
440 GO TO (500*(r=1)+700*(r=2)+
    5000*(r=3))
500 LET oq=-(ms*c)
510 IF oq<0 OR oq>200 THEN
    GO TO 550
520 PLOT oq+16,20: DRAW 0,150
530 LET sv=oq+15
540 LET ss=10: LET sz=0
545 GO SUB 2000
550 LET ow=-(mb*d)
560 IF ow<0 OR ow>150 THEN
    GO TO 600
570 PLOT 16,ow+20: DRAW 200,0
580 LET sv=232: LET sz=ow+8:
    LET ss=10
590 GO SUB 2000
595 PRINT AT 0,0;" "
600 GO TO 1500
700 INPUT "INCREMENTO X ?  ";ix
710 LET t=(md-ms)/ix
720 LET k=(200*ix)/(md-ms)
730 IF t-INT t<>0 THEN
    PRINT AT 21,0;"INCREMENTO
   X NON COMPATIBILE": PAUSE 3
   0: FOR h=0 TO 31: PRINT
   AT 21,h;" ": NEXT h:
    GO TO 700
800 INPUT "INCREMENTO Y ?  ";iy
810 LET p=(ma-mb)/iy
820 LET f=(150*iy)/(ma-mb)
830 IF p-INT p<>0 THEN
    PRINT AT 21,0;"INCREMENTO
   Y NON COMPATIBILE": PAUSE 3
   0: FOR h=0 TO 31: PRINT
   AT 21,h;" ": NEXT h:
    GO TO 730
900 REM reticolo
910 FOR j=0 TO t
920 PLOT k*j+16,20: DRAW 0,150
930 NEXT j
950 FOR j=0 TO p
960 PLOT 216,f*j+20: DRAW -200,
    0
970 NEXT j
1000 REM numeri ascisse
1005 LET flag=0
1010 FOR j=0 TO t
1020 LET sz=0: LET q=0
1030 LET v=ms+ix*j
1040 LET v$=STR$ (v)
1050 LET lv=LEN (v$): IF VAL (v$
    )>-0.1 AND VAL (v$)<0
     THEN  LET v=0: LET v$="0":
     LET lv=1
1060 IF VAL (v$)>0 AND VAL (v$)<
    0.1 THEN  LET v=0: LET v$="
    0": LET lv=1
1070 LET sv=k*j-6*lv/2+12
1080 IF sv<=flag+2 THEN  GO TO 1
    160
1090 IF v$(1 TO 1)="-" THEN
     LET sv=sv+3: LET ss=11:
     LET q=1: GO SUB 2000
1100 FOR i=1+q TO lv
1110 LET sv=sv+6
1120 IF v$(i TO i)="." THEN
     LET sv=sv-3: LET ss=12:
     GO SUB 2000: LET sv=sv-2:
     GO TO 1150
1130 LET ss=VAL (v$(i TO i)):
     IF ss=0 THEN  LET ss=10
1140 GO SUB 2000
1150 NEXT i
1155 LET flag=sv-2
1160 NEXT j
1180 PRINT AT 0,0;" "
1200 REM  numeri ordinate
1205 LET flag=1
1210 IF p>20 THEN  LET flag=2
1215 IF p>40 THEN  LET flag=4
1220 FOR j=0 TO p STEP flag
1230 LET q=0
1240 LET z=mb+iy*j
1250 LET z$=STR$ (z)
1260 LET lz=LEN (z$): IF VAL (z$
    )>-0.1 AND VAL (z$)<0
     THEN  LET z=0: LET z$="0":
     LET lz=1
1265 IF VAL (z$)>0 AND VAL (z$)<
    0.1 THEN  LET z=0: LET z$="
    0": LET lz=1
1270 LET sv=244-lz*6: LET sz=f*j
    +9
```

```
1280 IF lz>6 THEN GO TO 1360
1290 IF z$(1 TO 1)="-" THEN
  LET sv=sv+6: LET ss=11:
  LET q=1: GO SUB 2000
1300 FOR i=1+q TO lz
1310 LET sv=sv+6
1320 IF z$(i TO i)="." THEN
  LET ss=12: GO TO 1340
1330 LET ss=VAL (z$(i TO i)):
  IF ss=0 THEN  LET ss=10
1340 GO SUB 2000
1350 NEXT i
1360 NEXT j
1500 INPUT "VUOI CONTINUARE ?
 (S/N) ",e$
1510 IF e$<>"s" AND e$<>"S"
  AND e$<>"n" AND e$<>"N
 " THEN  BEEP .5,12: GO TO 1
 500
1520 IF e$="S" OR e$="s" THEN
  LET cont=1: GO TO 20
1530 GO TO 5000
2000 REM
2010 LET as=sv: LET or=9+sz
2020 FOR g=1 TO 40 STEP 2
2025 IF (n$(ss)(g TO g)=" ")
  THEN  LET g=40: GO TO 2200

2030 IF (n$(ss)(g TO g)="1")
  AND (as>=0) THEN  PLOT as,
 or
2040 LET s$=n$(ss)(g+1 TO g+1):
  LET il=VAL (s$): GO TO (21
 00+10*il)
2100 REM
2110 LET as=as+1: GO TO 2200
2120 LET or=or+1: GO TO 2200
2130 LET as=as-1: GO TO 2200
2140 LET or=or-1: GO TO 2200
2200 NEXT g
2220 RETURN
4000 REM matrice
4010 LET n$(1)="01011212121212l2
 "
4020 LET n$(2)="12120202120111l1
 04140314031404111111"
4030 LET n$(3)="02120202120111l1
 04140314011414031312"
4040 LET n$(4)="02121212011201l4
 14141411403130212"
4050 LET n$(5)="02120212121111l1
 14040414140313120202111l"
4060 LET n$(6)="02121212120111l1
 14030304111104141403131 2"
4070 LET n$(7)="02020202021111l1
 14141403140314031 2"
4080 LET n$(8)="02121202120111l1
 04140313140101141403131 2"
4090 LET n$(9)="12020212120111l1
 04141413131401011403131 2"
4100 LET n$(10)="0212121212012111l1
 104141414140313 13"
4110 LET n$(11)="02020201111111"

4120 LET n$(12)="010111"
4150 RETURN
5000 STOP
9000 CLS : PRINT AT 5,2; FLASH 1
 ;"F E R M A   IL   N A S T
  R O ";AT 16,9; FLASH 0;"Pr
 emi un tasto": FOR h=0 TO 2
 4: BEEP .1,h: NEXT h:
  PAUSE 0
9005 CLS
9010 FOR x=0 TO 255 STEP 8:
  PLOT x,0: DRAW 0,167:
  NEXT x
9015 FOR y=0 TO 175 STEP 8:
  PLOT 0,y: DRAW 247,0:
  NEXT y
9020 FOR t=16 TO 96: PLOT 24,t:
  DRAW 7,0: NEXT t
9025 FOR t=96 TO 88 STEP -1:
  PLOT 32,t: DRAW 39,0:
  PLOT 32,t-40: DRAW 39,0:
  NEXT t
9030 FOR t=56 TO 88: PLOT 64,t:
  DRAW 7,0: NEXT t


9035 FOR t=48 TO 120: PLOT 96,t:
  DRAW 7,0: NEXT t
9040 FOR t=48 TO 96: PLOT 128,t:
  DRAW 7,0: PLOT 168,t:
  DRAW 7,0: NEXT t
9045 FOR t=96 TO 88 STEP -1:
  PLOT 136,t: DRAW 39,0:
  PLOT 136,t-40: DRAW 39,0:
  NEXT t


9050 FOR t=48 TO 120: PLOT 200,t
 : DRAW 7,0: NEXT t
9055 FOR t=112 TO 104 STEP -1:
  PLOT 192,t: DRAW 32,0:
  NEXT t
9060 FOR x=-10 TO 10 STEP 0.2:
  INK 2: LET y=SIN (x):
  PLOT x*10+128,y*10+151:
  NEXT x
9070 PAUSE 150: CLS : INK 0
9080 RUN
```

King Kong

di Nicholas Wyre
trad. e adatt. Marcello Spero

In quanto tempo potete salvare la vostra adorata principessa?

Scopo di questo gioco è liberare una principessa dalla prigionia del malvagio King Kong, il gorilla gigante. Per riuscire in questa impresa dovrete superare tre fasi.

1 - Arrivare alla prigione in cui è segregata la principessa salendo le scale fino all'ultimo livello, con il rischio di cadere in uno dei buchi o di finire calpestati da uno dei gorilla di guardia. Una volta giunti al livello più elevato avrete due alternative per raggiungere la principessa: sgattaiolare fra le zampe di King nel momento in cui c'è spazio o salire sul barile rotolante e in entrambi i casi correte il rischio di fare una brutta fine: essere presi da King o cadere dal barile.

2 - Non appena liberata, la principessa correrà verso l'ascensore e vi entrerà. A questo punto dovrete scendere nuovamente fino al primo livello e premere il bottone posto nell'angolo sinistro in basso dello schermo, per far partire l'ascensore.

3 - Infine dovrete togliere tutti i bulloni che tengono insieme il pavimento dei vari livelli, passandoci sopra. Quando avrete fatto saltare l'ultimo bullone King Kong rovinerà verso il basso, e sarà la sua fine. Attenti, però: se durante un'altra fase del gioco avrete inavvertitamente tolto uno o più bulloni non potrete completare la vostra prova; quindi, attenzione!

Consigli per l'uso

Trascrivete tutto il programma e quindi date RUN 1170: vedrete il solito messaggio "start tape, then press any key" che precede ogni registrazione; preparate il registratore e salvate il programma su cassetta. Una volta effettuata questa operazione il computer passerà automaticamente alla creazione dei caratteri grafici; quindi vi chiederà di nuovo di preparare il registratore: fatelo partire in modo da salvare la grafica subito di seguito al programma. A questo punto tutto è pronto: basterà infatti fare LOAD "KING-KONG" per ottenere l'autostart del programma, che immediatamente caricherà la grafica.

Per i proprietari della versione 16K, invece, la procedura sarà leggermente diversa. Il programma, infatti occuperebbe tutta la memoria disponibile, senza lasciare spazio per le variabili, ed il funzionamento sarebbe quindi impossibile. Per aggirare questo ostacolo fate così:

1 - trascrivere il programma solo fino alla linea 1170
2 - date RUN 1170 e salvatelo su cassetta
3 - date NEW
4 - trascrivete la restante parte del programma, cioè le linee 1180-1290
5 - date RUN 1180 e salvate la grafica, subito al programma
6 - date NEW per eliminare questa parte del programma, che in effetti è inutile.

A questo punto potete caricare il programma, che partirà da solo caricandosi la sua grafica.

Infine una possibile modifica per variare la difficoltà di gioco. La "cattiveria" con cui i gorilla vi inseguono è controllata dal numero in linea 160, che normalmente è 0.75. Se lo portate ad 1 i gorilla non vi inseguiranno per niente, muovendovi a caso, mentre se ne diminuite il valore sarete inseguiti più spesso.

Controlli

I tasti che guidano il vostro personaggio attraverso lo schermo sono:

4 - per saltare a sinistra
5 - per correre a sinistra
6 - per scendere una scala
7 - per salire una scala
8 - per correre a destra
9 - per saltare a destra

Descrizione del programma

Il programma comprende due parti: il programma vero e proprio e la routine di caricamento per i caratteri grafici. Qui di seguito trovate una breve spiegazione delle varie operazioni compiute dal programma.

Linee	Commento
10-30	carica i caratteri grafici
40-50	inizializza il contatempo
60-140	ciclo principale del programma
100-130	disegna King Kong
140-270	muove i gorilla ed il barile
190-320, 370,380	muove il vostro personaggio
330-360	controlla se avete urtato qualcosa
280,390	controlla se avete raggiunto la principessa o premuto il bottone
420-440	vi avvisa che siete morti
450-470	vi chiede se volete giocare ancora
480-520	cadete in un buco
530-550	siete morti
560-610	avete spezzato la corda che sostiene il barile
620-650	avete pestato un bullone
660-670	la principessa corre all'ascensore
780-800	saltate a sinistra
810-830	saltate a destra
840-870	cade il pavimento di un livello
880-900	l'ascensore sale con la principessa
910-920	salta l'ultimo bullone, e con lui precipita King Kong
930-970	volete giocare ancora?
980-990	vi viene comunicato che avete realizzato il miglior tempo
1000-1160	creazioni dello scenario



King Kong
versione per ZX Spectrum 16K/48K

```
 10 REM       "KING-KONG"
       di       NILCHOLAS WYR
   E
 20 LOAD "GRAFICA"CODE USR "A",
   18*8: BORDER 2: INK 3:
    PAPER 5: CLS : LET H=9999:
    LET N$="NICHOLAS WYRE"
 30 LET X=0: LET E=1: LET F=-1:
    LET C=1: LET V=21: LET G=5
   00: LET M=21: LET N=2:
    LET E=1: LET T=0: LET E$="
   G": GO SUB 1000: LET B=2:
    LET D=1
 40 POKE 23672,0: POKE 23673,0:
    POKE 23674,0
 50 DEF FN S()=INT ((PEEK 23672
   +PEEK 23673*256+PEEK 23674*
   65535)/50)
 60 PRINT AT M,N; INK 1; OVER 1
   ;E$: BEEP .05,10: PRINT
   AT M,N; OVER 1; INK 1;E$
 70 PRINT AT 2,B; INK 1;" R ";
   AT 1,B;" : "
 80 PRINT AT 9,C;" L L L ";
   AT 13,V;" L L L ";AT 17,C
   ;" L L L ";AT 21,V;" L L
   L "
 90 LET Z$=INKEY$
100 IF RND<.5 THEN  PRINT AT 3,
   19;"{G3}"; INVERSE 1;"#";
    INVERSE 0;"{SG7}": GO TO 1
   20
110 PRINT AT 3,19;"A{SG8}C"
120 IF RND<.5 THEN  PRINT AT 5,
   19;"D E": GO TO 140
130 PRINT AT 5,19;"ED"
140 IF B=2 THEN  LET D=1
150 IF B=6 THEN  LET D=-1
160 IF RND<.75 THEN  GO TO 220
170 IF V>2 AND V>N THEN  LET V=
   V-1
180 IF V<21 AND V<N THEN  LET V
   =V+1
190 IF C<21 AND C<N THEN  LET C
   =C+1
200 IF C>2 AND C>N THEN  LET C=
   C-1
210 GO TO 270
220 LET C=C+E: LET V=V+F
230 IF C>21 THEN  LET E=-1
240 IF C<3 THEN  LET E=1
250 IF V<3 THEN  LET F=1
260 IF V>21 THEN  LET F=-1
270 LET B=B+D
280 IF M=2 AND N=11 AND X=0
    THEN  GO SUB 660
290 IF Z$="7" AND ATTR (M-1,N)=
   41 THEN  LET E$="N": LET M=
   M-1
300 IF Z$="6" AND ATTR (M+1,N)=
   41 THEN  LET E$="N": LET M=
   M+1
310 IF Z$="5" AND N>0 THEN
    LET N=N-1: LET E$="K"
320 IF Z$="8" AND N<>31
    THEN  LET E$="H": LET N=N+
   1
330 IF SCREEN$ (M,N)<>" "
    AND ATTR (M,N)=40 THEN
    GO TO 420
340 IF SCREEN$ (M,N)="_" THEN
    GO TO 560
350 IF SCREEN$ (M+1,N)=" "
    AND M<>21 THEN  GO TO 48
   0
360 IF ATTR (M+1,N)=107 THEN
    GO SUB 620
370 IF Z$="4" AND N>1 THEN
    GO SUB 780
380 IF Z$="9" AND N<30 THEN
    GO SUB 810
390 IF M=21 AND N=0 AND X=1
    THEN  GO SUB 880
400 BEEP .01,1
410 GO TO 60
420 PRINT AT M,N; FLASH 1;"O";
   AT 10,13;"CRUNCH"
430 FOR N=0 TO 7: FOR X=0 TO 7:
    BEEP .05,N: BORDER RND*7:
    NEXT X: NEXT N
440 PAPER 3: INK 7: CLS :
    FLASH 1: PRINT AT 8,5;(" K
   ING KONG TI HA PRESO "
    AND M=5);AT 8,5;("SEI STAT
   O SCHIACCIATO " AND M<>5
   )
450 PRINT AT 18,1;"PREMI UN TAS
   TO"''" PER GIOCARE ANCORA";
   AT 14,1;"MIGLIOR TEMPO = ";
   H''"di ";n$
460 IF INKEY$="" THEN  BEEP .01
   ,RND*25: BORDER RND*7:
    GO TO 460
470 FLASH 0: BORDER 2: INK 3:
    PAPER 5: CLS : GO TO 30
480 IF SCREEN$ (M,N)<>" "
    THEN  LET M=N-1: GO TO 530
```

```
490 PRINT AT M,N;"G";AT M-1,N;"
    "
500 IF M=21 THEN GO TO 530
510 LET M=M+1: BEEP .03,M
520 GO TO 480
530 PRINT AT M,N; FLASH 1;"O";
    AT 10,13;"SPLAT!"
540 FOR N=0 TO 7: FOR S=7 TO 0
     STEP -1: BEEP .01,S*N:
     BORDER S: NEXT S: NEXT N
550 PAPER 4: INK 1: CLS :
     FLASH 1: PRINT AT 8,9;"SEI
     CADUTO";AT 10,4;" LA PARTI
    TA E' FINITA": GO TO 450
560 PRINT AT 1,2;"-         -";AT 2
    ,2;"|        |";AT 3,2;"|
    |";AT 4,2;"R": LET N=N+(1
     AND N=2)-(1 AND N=7)
570 FOR A=2 TO 5: PRINT AT A-1,
    N;" "
580 BEEP .04,M
590 PRINT AT A,N;"G": NEXT A
600 LET M=5
610 GO TO 530
620 LET T=T+1: PRINT AT M+1,N;
    "-"
630 IF SCREEN$ (M+1,27)="-"
     AND N=13 THEN GO SUB 840
640 IF SCREEN$ (M+1,13)="-"
     AND N=27 THEN GO SUB 840
650 BEEP .01,50: RETURN
660 LET X=1
670 PRINT AT 0,8; FLASH 1;"MMMM
    MM": BEEP 1,10
680 FOR A=11 TO 13: PRINT AT 1,
    A;" P";AT 2,A;" Q": BEEP .
    05,A: NEXT A
690 INK 1
700 PRINT AT 2,14;"F": BEEP .05
    ,50
710 FOR A=14 TO 25: PRINT AT 0,
    A;" P";AT 1,A;"{G3}Q":
     BEEP .01,A: NEXT A
720 PRINT AT 0,13;" F";AT 1,14;
    "F"
730 FOR A=0 TO 3: PRINT AT A,2
    6;"F";AT A+1,26;"P";AT A+2,
    26;"Q": BEEP .05,A: NEXT A
740 PRINT AT 4,26;"FP";AT 5,26;
    "FQ": BEEP .0,50
750 FOR A=27 TO 29: PRINT AT 4
    ,A;" P";AT 5,A;" Q": BEEP .
    05,A: NEXT A
760 INK 1: PRINT AT 4,29;"{G5}"
    ;AT 5,29;"{G5}";AT 4,30;
     FLASH 1;"P";AT 5,30;"Q"
770 FLASH 0: RETURN
780 PRINT AT M-1,N-1; OVER 1;"J
    ": BEEP .05,M: PRINT AT M-1
    ,N-1; OVER 1;"J": BEEP .05,
    M
790 IF SCREEN$ (M,N-2)<>" "
     AND ATTR (M,N-2)=40 THEN
     GO TO 420
800 LET N=N-2: RETURN
810 PRINT AT M-1,N+1; OVER 1;"I
    ";: BEEP .05,M: PRINT AT M-
    1,N+1; OVER 1;"I": BEEP .05
    ,M
820 IF SCREEN$ (M,N+2)<>" "
     AND ATTR (M,N+2)=40 THEN
     GO TO 420
830 LET N=N+2: RETURN
840 IF T=8 AND X=2 THEN
     GO TO 910
850 FOR Q=M+1 TO M+4
860 BEEP .01,Q: PRINT AT Q-1,14
    ;"            ";AT Q,14;
     INVERSE 1;"############
    ": NEXT Q
870 RETURN
880 LET X=2: PRINT AT 4,30;"
    {SG8}";AT 5,30;"{SG8}";AT 3
    ,30;"{SG8}"
890 BEEP 1,0
900 FOR A=5 TO 2 STEP -1:
     PRINT AT A,29;"   ";AT A-3
    ,29;"{G5}{SG8}{SG5}":
     BEEP .1,A: NEXT A: PRINT
    AT 2,29;"   ": BEEP .1,10:
     PRINT AT 1,29;"   ":
     BEEP .1,15: PRINT AT 0,29;
    "   ": BEEP 1,20: RETURN
910 LET S=FN S(): PRINT AT 2,20
    ;" ";AT 3,19;"A"; INVERSE 1
    ;"O"; INVERSE 0;"CGULP!":
     BEEP .5,2: PRINT AT 3,19;"
         ";AT 4,19;
     INVERSE 1;"@  ";AT 5,20;
     "D": BEEP .5,3: PRINT AT 4,
     19;"  ";AT 5,20;" ";AT 5,1
     9;"B"; INVERSE 1;"O.";
     INVERSE 0;"E": BEEP .5,3
920 FOR Q=6 TO 21: PRINT AT Q,1
    4; INVERSE 1;"############
    "; INVERSE 0;AT Q-2,19;"
     ";AT Q-1,14;"   ";
     INVERSE 1;"BC"; INVERSE 0
    ;"E   ": BEEP .05,Q:
     NEXT Q: FOR J=-50 TO 50:
     BEEP .03,J: NEXT J
```

```
930 PAUSE 100: CLS : PRINT AT 2
    ,5;"HAI ELIMINATO KONG
       E SALVATO LA PRINCIPE
    SSA"'"      TEMPO = ";S
940 IF S<H THEN  GO SUB 980
950 PRINT AT 13,5;"MIGLIOR TEMP
    O = ";H'"   DEL PRODE ";N$

960 PRINT AT 18,0; FLASH 1;"
        PREMI UN TASTO
           PER GIOCARE ANCORA
          ": BORDER RND*7:
     BEEP .01,RND*25: IF
    INKEY$="" THEN  GO TO 960
970: BEEP 1,8: CLS : GO TO 30
980 LET H=S: INPUT "HAI REALIZZ
    ATO IL MIGLIOR TEMPO
    SCRIVI IL TUO NOME, PREGO "
    ;N$
990 RETURN
1000 FOR Q=5 TO 21: PRINT AT Q,0
    ; INK 1;"F";AT Q,31;"F":
     NEXT Q
1010 PRINT AT 6,1; INK 0;
     INVERSE 1;"###############
    ################";AT 10,0;"
    #{SG8}##{SG8}#{SG8}##{SG8}#
    {SG8}#################{SG8}
    #";AT 14,1;"###{SG8}####
    {SG8}#{SG8}################
    #{SG8}##";AT 18,0;"####
    {SG8}##{SG8}###{SG8}#######
    ##########{SG8}#"
1020 PRINT AT 11,0; OVER 1;
     INK 2;"{"'"{"
1030 FOR Q=7 TO 8: PRINT AT Q,31
    ; OVER 1; INK 2;"}";AT Q+8,
    31;"}": NEXT Q
1040 PRINT AT 19,0; PAPER 4;"@"'
    "@"'"@"
1050 INK 1:: PRINT AT 1,2;"
    "
1060 PRINT AT 4,26;"F";AT 5,26;"
    F";AT 4,1;"F";AT 5,1;"F"
1070 FOR Q=0 TO 3: PRINT AT Q,1;
    "F";AT Q,9;"F";AT Q,14;"F";
    AT Q,26;"F": NEXT Q

1080 PRINT AT 0,9; FLASH 1;"AIUT
    O!"
1090 PRINT AT 3,9; INK 0;
     INVERSE 1;"######";AT 1,15
    ; INK 1;"{11SG3}"
1100 PRINT AT 1,11;"P";AT 2,11;"
    Q"
1110 INK 1
1120 PRINT AT 3,29;"{G5}{G3}
    {SG5}";AT 4,29;"{G5} {SG5}"
    ;AT 5,29;"{G5}{SG5}";
     INK 6;AT 0,30;"|";AT 1,30;
    "|";AT 2,30;"|"
1130 INK 0: PRINT AT 2,20;"B";
    AT 3,19;"A{SG8}C";AT 4,20;"
    {SG8}";AT 5,19;"D E"
1140 INK 3: BRIGHT 1: PRINT AT 6
    ,13;"+";AT 6,27;"+";AT 10,1
    3;"+";AT 10,27;"+";AT 14,13
    ;"+";AT 14,27;"+";AT 18,13;
    "+";AT 18,27;"+"
1150 BRIGHT 0: INK 0
1160 RETURN
1170 SAVE "KING-KONG" LINE 10
1180 FOR A=144 TO 161
1190 FOR U=0 TO 7
1200  READ I
1210 POKE USR (CHR$ A)+U,I
1220 NEXT U
1230 NEXT A
1240 DATA 0,0,16,8,7,31,7,24,126
    ,219,219,255,27,103,195,126
    ,0,248,252,252,28,60,20,16,
    31,127,127,112,60,30,6,60
1250 DATA 254,254,254,2,2,0,0,0,
    66,66,66,126,66,66,66,66,56
    ,40,16,254,186,186,40,108,2
    4,29,10,60,88,30,98,65
1260 DATA 48,58,20,24,24,4,56,0,
    12,72,40,24,24,32,28,0,24,1
    84,80,60,58,120,70,130,153,
    189,153,255,60,60,102,195
1270 DATA 0,34,119,127,62,28,8,8
    ,28,28,40,0,28,28,32,0,0,0,
    0,0,16,97,221,255,129,153,1
    89,189,153,255,24,24,24,60,
    60,126,126,255,255,219
1280 DATA 56,66,129,165,165,129,
    66,56
1290 SAVE "GRAFICA"CODE USR "A",
    18*8
```

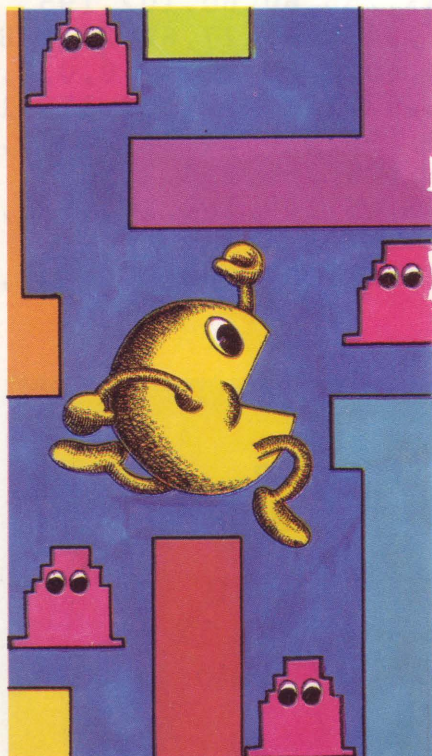
Spedite i vostri programmi migliori a:
SUPERSINC
Via Rosellini, 12
20124 Milano

Chomp Chomp

di Marcello Spero

Ovvero, come realizzarsi in casa il proprio «Pac»...

Il gioco, una versione del classico di tutte le sale giochi, coinvolge un piccolo esserino affamato di puntini, inseguito da due fantasmi affamati di lui. Ai quattro angoli del labirinto, per fortuna, ci sono le famose pillole di energia che permetteranno al nostro eroe di mettersi alla caccia dei fantasmi: ma in fretta! L'effetto delle pillole è infatti limitato, e da un momento all'altro i mostri potrebbero rivolgerrsi nuovamente contro di lui. L'effetto delle pillole è reso evidente dal cambiamento di colore dei fantasmi, che diventano verdi e lampeggianti. Di tanto in tanto apparirà una ciliegia, proprio sotto la tana dei mostri: mangiata, darà 10 punti. Ogni puntino, invece, vale un punto mentre le pillole e i mostri fruttano 10 punti ciascuno. Realizzare punteggi elevati è semplicissimo: basta sopravvivere abbastanza!



Descrizione delle variabili

s	punteggio
hs	punteggio record
t	punteggio corrispondente ad uno schermo completo
b$	il labirinto
cont	se è meno di 30 i mostri possono essere mangiati
vite	il numero di vite rimaste
x, y	la vostra posizione
a$	la vostra immagine
gx, gy	la posizione del primo fantasma
gx1, gy1	la posizione del secondo fantasma
c	se è -1 i mostri fuggono da voi, se è 1 vi cercano
d$	ciò che c'è sotto il primo fantasma
e$	ciò che c'è sotto il secondo fantasma
a	livello di gioco
n	viene usata per vari scopi

Descrizione del programma

Linee	Commento
1-2	crea i caratteri grafici
3-4	inizializza alcune variabili
10-50	i dati per la grafica
60-260	crea in b$ il labirinto
270	disegna il labirinto
280-290	inizializza altre variabili
295	stampa il punteggio record
300-320	visualizza voi e i fantasmi
330	decide se visualizzare la ciliegia
340-370	legge la tastiera e va alla subroutine corrispondente
375	stampa il punteggio
380-400	controlla cosa c'è nella vostra posizione, e aumenta il punteggio di conseguenza
410	visualizza il vostro personaggio
415	se lo schermo è stato completato va alla linea **5000**
420	visualizza ciò che era sotto i fantasmi
425-490	muove in maniera appropriata i fantasmi
500	incrementa cont
510-535	visualizza i fantasmi, in verde lampeggiante se cont è minore di 30, o ciano e magenta se è maggiore
540-545	se occupate la medesima posizione di un fantasma va alla linea 9000
550	aggiorna d$ e$
1000-1030	visualizza la ciliegia
1500-1540	muove il vostro personaggio verso destra
1600-1640	muove il vostro personaggio verso sinistra
1700-1730	muove il vostro personaggio verso l'alto
1800-1830	muove il vostro personaggio verso il basso
2000-2050	se mangiate una pillola di energia le variabili vengono aggiornate di conseguenza
5000-5070	routine grafica fra uno schermo e l'altro
7000-7090	seleziona il livello di gioco
8010-9000	avete mangiato un fantasma
9000-9030	siete stati mangiati, e le variabili vengono aggiornate di conseguenza
9500-9530	la partita è finita, e le variabili vengono preparate per la partita successiva

Chomp Chomp
versione per ZX Spectrum 16K/48K

```
  1 RESTORE 0: FOR a=144 TO 164

  2 FOR b=0 TO 7: READ c:
     POKE USR CHR$ a+b,c:
     NEXT b: NEXT a
  3 LET s=0: LET hs=0: LET t=18
    0: DIM b$(21,21): LET cont=
    30
  4 LET vite=3
  5 GO SUB 7000
 10 DATA 0,255,0,0,0,0,0,0,0,0,
    0,0,0,0,255,0,0,255,0,0,0,0
    ,255,0,0,248,4,2,2,2,2,2
 20 DATA 0,31,32,64,64,64,64,64
    ,0,63,64,128,128,64,63,0,0,
    252,2,1,1,2,252,0
 30 DATA 2,2,2,2,2,4,248,0,64,6
    4,64,64,64,32,31,0,2,2,2,2,
    2,2,2,2
 40 DATA 0,24,36,66,66,66,66,66
    ,66,66,66,66,66,36,24,0,64,
    64,64,64,64,64,64,64,66,66,
    66,66,66,66,66,66,0,0,16,56
    ,124,56,16,0
 50 DATA 24,60,30,15,30,60,24,0
    ,0,0,34,119,127,62,28,8,0,2
    4,60,120,240,120,60,24,16,5
    6,124,254,238,68,0,0,56,126
    ,90,219,255,255,255,147,2,6
    ,10,20,36,68,238,68
 55 PAPER 1: BORDER 1: CLS :
     INK 7: PAPER 0
 60 LET b$(1)="EAAAAAAAAANAAAAA
    AAAAD"
 70 LET b$(2)="M.........N.....
    ....J"
 80 LET b$(3)="M.EAD.EAD.N.EAD.
    EAD.J"
 90 LET b$(4)="MOM J.M J.N.M J.
    M JOJ"
100 LET b$(5)="M.IBH.IBH.L.IBH.
    IBH.J"
110 LET b$(6)="M...............
    ....J"
120 LET b$(7)="M.FCG.K.FCACG.K.
    FCG.J"
130 LET b$(8)="M.....N...N...N.
    ....J"
140 LET b$(9)="IBBBB.MCG.L.FCJ.
    BBBBH"
150 LET b$(10)="    J.N......N
    .M    "
160 LET b$(11)="BBBBH.L.E---D.L
    .IBBBB"
170 LET b$(12)="/.......M   J..
    .....\"
180 LET b$(13)="AAAAD.K.I---H.K
    .EAAAA"
190 LET b$(14)="    J.N...  ..N
    .M    "
200 LET b$(15)="BBBBH.L.FCACG.L
    .IBBBB"
210 LET b$(16)="M.........N....
    ....J"
220 LET b$(17)="MOFCD.FCG.L.FCG
    .ECGOJ"
230 LET b$(18)="M...N..........
    .N...J"
240 LET b$(19)="AAD.L.FCCCCCCCG
    .L.EAA"
250 LET b$(20)="  J............
    ...M  "
260 LET b$(21)="   AAAAAAAAAAAA
    AAA   "
270 FOR n=1 TO 21: PRINT AT n,1
    ;b$(n): BEEP .05,n+15:
     NEXT n
280 LET y=14: LET x=12: LET a$=
    "R"
290 LET gx=12: LET gy=12: LET g
    x1=12: LET gy1=11: LET c=1:
     LET d$=" ": LET e$=" "
295 PRINT AT 0,19;"RECORD =";hs
300 PRINT AT gx,gy; INK 3;"T"
310 PRINT AT gx1,gy1; INK 5;"T"

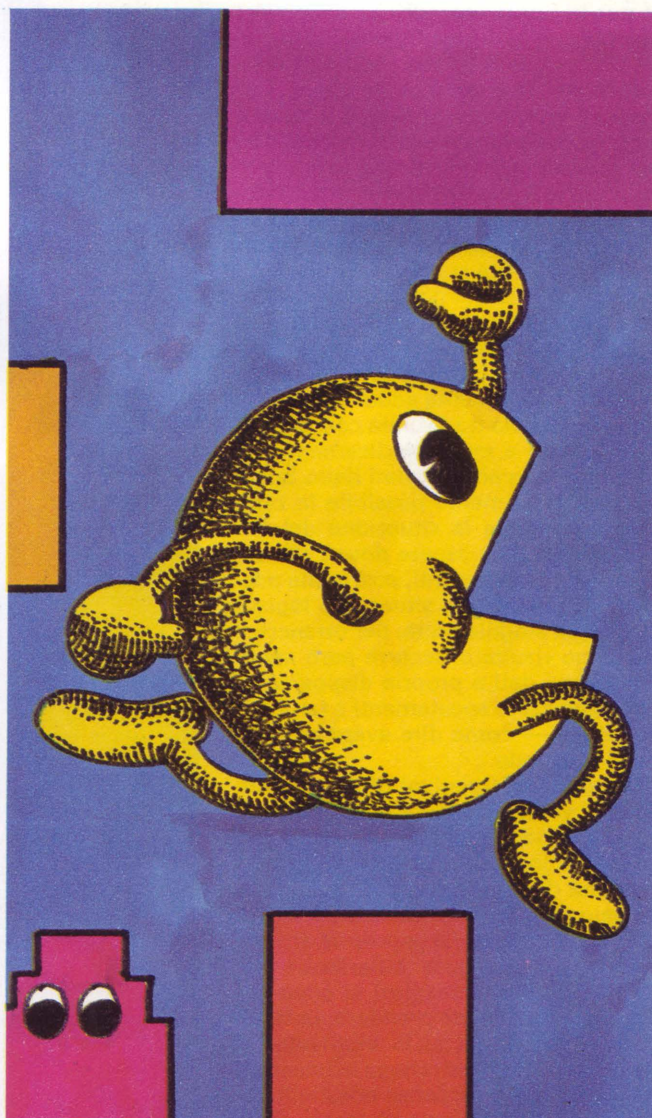
320 PRINT AT y,x; INK 6;a$
330 IF INT (RND*100)=0 THEN
     GO SUB 1000
340 IF INKEY$="8" THEN
     GO SUB 1500
350 IF INKEY$="5" THEN
     GO SUB 1600
360 IF INKEY$="7" THEN
     GO SUB 1700
370 IF INKEY$="6" THEN
     GO SUB 1800
380 IF b$(y,x)="." THEN LET s=
    s+1: BEEP .05,25
390 IF b$(y,x)="O" THEN LET s=
    s+10: BEEP .1,25: BEEP .1,3
    9: GO SUB 2000
400 IF b$(y,x)="U" THEN LET s=
    s+10: LET t=t+10: BEEP .1,2
    8: BEEP .1,36
401 PRINT AT 0,0;"VITE =";vite;
    " PUNTI =";s
405 LET b$(y,x)=" "
```

```
410 PRINT AT y,x; INK 6;a$
415 IF s=t THEN  GO TO 5000
420 PRINT AT gx,gy;d$;AT gxl,gy
    l;e$
425 IF RND<a THEN  GO TO 440
430 IF gy<x THEN  LET d=CODE b$
    (gx,gy+c): IF d<>47 AND d
    <>92 AND d<144 OR d>157
     THEN  LET gy=gy+c: GO TO 4
    70
435 IF RND<a THEN  GO TO 450
440 IF gy>x THEN  LET d=CODE b$
    (gx,gy-c): IF d<>47 AND d
    <>92 AND d<144 OR d>157
     THEN  LET gy=gy-c: GO TO 4
    70
445 IF RND<a THEN  GO TO 460
450 IF gx<y THEN  LET d=CODE b$
    (gx+c,gy): IF d<>47 AND d
    <>92 AND d<144 OR d>157
     THEN  LET gx=gx+c: GO TO 4
    70
455 IF RND<a THEN  GO TO 470
460 IF gx>y THEN  LET d=CODE b$
    (gx-c,gy): IF d<>47 AND d
    <>92 AND d<144 OR d>157
     THEN  LET gx=gx-c
465 IF RND<a THEN  GO TO 475
470 IF gyl<x THEN  LET d=CODE b
    $(gxl,gyl+c): IF d<>47
     AND d<>92 AND d<144 OR d
    >157 THEN  LET gyl=gyl+c:
     GO TO 500
472 IF RND<a THEN  GO TO 480
475 IF gyl>x THEN  LET d=CODE b
    $(gxl,gyl-c): IF d<>47
     AND d<>92 AND d<144 OR d
    >157 THEN  LET gyl=gyl-c:
     GO TO 500
477 IF RND<a THEN  GO TO 490
480 IF gxl<y THEN  LET d=CODE b
    $(gxl+c,gyl): IF d<>47
     AND d<>92 AND d<144 OR d
    >157 THEN  LET gxl=gxl+c:
     GO TO 500
485 IF RND<a THEN  GO TO 500
490 IF gxl>y THEN  LET d=CODE b
    $(gxl-c,gyl): IF d<>47
     AND d<>92 AND d<144 OR d
    >157 THEN  LET gxl=gxl-c
500 LET cont=cont+1
510 IF cont>=30 THEN  LET c=1
520 IF c=-1 THEN  PRINT AT gx,g
    y; INK 4; FLASH 1;"T"
525 IF c=1 THEN  PRINT AT gx,gy
    ; INK 3;"T"
530 IF c=-1 THEN  PRINT AT gxl,
    gyl; INK 4; FLASH 1;"T"
535 IF c=1 THEN  PRINT AT gxl,g
    yl; INK 5;"T"
540 IF gx=y AND gy=x THEN
     GO TO 9000
545 IF gxl=y AND gyl=x THEN
     GO TO 9000
550 FLASH 0: LET d$=b$(gx,gy):
     LET e$=b$(gxl,gyl):
     GO TO 330
1000 IF b$(14,11)="U" THEN
     RETURN
1010 PRINT AT 14,11; INK 2;"U":
     BEEP .5,15: IF b$(14,11)="
     ." THEN  LET t=t-1
1020 LET b$(14,11)="U"
1030 RETURN
1500 PRINT AT y,x;" "
1505 IF b$(y,x+1)="." THEN
     LET x=x+1: GO TO 1530
1510 IF b$(y,x+1)=" " THEN
     LET x=x+1: GO TO 1530
1520 IF b$(y,x+1)="U" THEN
     LET x=x+1
1525 IF b$(y,x+1)="\" THEN
     LET x=2
1530 LET a$="R"
1540 RETURN
1600 PRINT AT y,x;" "
1605 IF b$(y,x-1)="." THEN
     LET x=x-1: GO TO 1630
1610 IF b$(y,x-1)=" " THEN
     LET x=x-1: GO TO 1630
1615 IF b$(y,x-1)="O" THEN
     LET x=x-1: GO TO 1630
1620 IF b$(y,x-1)="/" THEN
     LET x=20: GO TO 1630
1625 IF b$(y,x-1)="U" THEN
     LET x=x-1
1630 LET a$="P"
1640 RETURN
1700 PRINT AT y,x;" "
1705 IF b$(y-1,x)="." THEN
     LET y=y-1: GO TO 1725
1710 IF b$(y-1,x)=" " THEN
     LET y=y-1: GO TO 1725
1720 IF b$(y-1,x)="O" THEN
     LET y=y-1: GO TO 1725
1725 LET a$="Q"
1730 RETURN
1800 PRINT AT y,x;" "
1805 IF b$(y+1,x)="." THEN
     LET y=y+1: GO TO 1825
1810 IF b$(y+1,x)="O" THEN
     LET y=y+1: GO TO 1825
```

```
1815 IF b$(y+1,x)=" " THEN
      LET y=y+1
1825 LET a$="S"
1830 RETURN
2000 LET c=-1
2010 LET b$(y,x)=" "
2020 PRINT AT y,x;a$
2030 LET t=t+10
2040 LET cont=0
2050 RETURN
3000 IF y>d THEN  LET d=d+g
3010 IF x>f THEN  LET f=f+h
3020 IF x<f THEN  LET f=f-h
3030 IF x<d THEN  LET d=d-g
3040 RETURN
4010 LET d=gx: LET f=gy:
      RETURN
4020 LET d=gx1: LET f=gy1
4030 RETURN
5000 CLS : PRINT AT 10,31;"O"
5010 FOR n=1 TO 27
5020 PRINT AT 10,n; INK 5;" T  "
     ; INK 7;"R"
5030 BEEP .05,n: NEXT n
5040 FOR n=27 TO 1 STEP -1:
      PRINT AT 10,n; FLASH 1;
      INK 4;"T"; FLASH 0; INK 7;
     " P "


5050 BEEP .05,n: NEXT n
5060 LET t=t+180
5070 GO TO 60
6010 LET gx=d: LET gy=f:
      RETURN
6020 LET gx1=d: LET gy1=f:
      RETURN
7000 PRINT AT 0,10;"CHOMP CHOMP"
     ;TAB 9;"{G1}{11G3}{G2}"
7010 PRINT AT 9,0;"AAAAAAAAAAAAA
     AAAAAAAAAAAAAAAAAAA"
7020 PRINT AT 11,4;"SCEGLI IL LI
     VELLO (1-5)             "
7030 PRINT TAB 5;"(5 e' il piu'
     facile)"
7040 PRINT AT 14,0;"AAAAAAAAAAAA
     AAAAAAAAAAAAAAAAAAAA"
7060 LET a=CODE INKEY$-49
7070 IF a>5 OR a<0 THEN  GO TO 7
     060
7080 LET a=a+1: LET a=a/10
7090 RETURN
8000 IF gy=x AND gx=y THEN
      LET s=s+10: LET t=t+10:
      LET gy=12: LET gx=12:
      BEEP .05,20: BEEP .05,10:
      BEEP .1,10: GO TO 550
8010 IF gy1=x AND gx1=y THEN
      LET s=s+10: LET t=t+10:
      LET gy1=11: LET gx1=12:
      BEEP .05,20: BEEP .05,10:
      BEEP .1,10: GO TO 550
9000 IF cont<=30 THEN  GO TO 800
     0
9005 LET vite=vite-1
9010 PRINT AT y,x; FLASH 1;a$:
      FOR n=50 TO 0 STEP -1:
      BEEP .05,n: NEXT n
9020 IF vite<=0 THEN  GO TO 9500

9030 GO TO 270
9500 IF hs<s THEN  LET hs=s
9510 PRINT AT 8,10;"PREMI";AT 10
     ,0;"UN TASTO PER RICOMINCIA
     RE"
9520 IF INKEY$="" THEN  GO TO 95
     20
9530 PAPER 1: LET s=0: CLS :
      PAPER 0: LET t=180: LET vi
     te=3: GO TO 60
```



R·U·N

Recensioni Software

a cura di
Lucio Bragagnolo

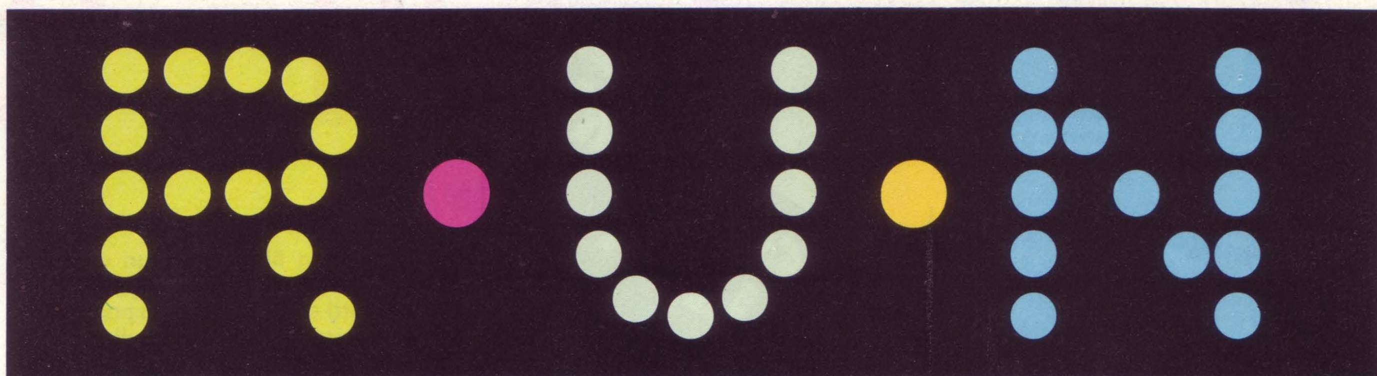
Stonkers

Avete mai affrontato in campo aperto il vostro computer?

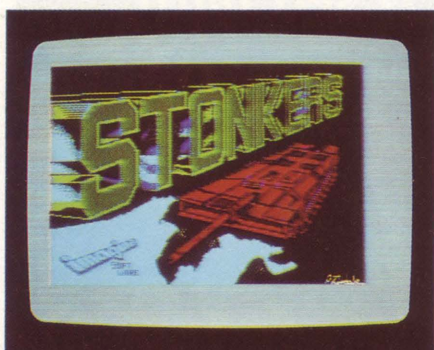
Si tratta di un tipico boardgame (una simulazione militare a tavolino) adattato per lo Spectrum 48K. Lo scopo del gioco è quello di sconfiggere il computer distruggendo le sue truppe o penetrando nel suo quartier generale. Ciò avviene individuando su una mappa a piccola scala la zona delle operazioni, per poi ingrandire la stessa e muovere le proprie unità (divisioni corazzate, fanteria e artiglieria) come si desidera. Contemporaneamente il calcolatore svilupperà la sua strategia e gestirà gli scontri tra le unità nemiche in contatto. Non è finita qui, però: le nostre truppe hanno necessità di essere rifornite, e dovremo provvedere alla bisogna con quattro unità addette, a cui faremo fare spola tra le unità combattenti e il porto a cui periodicamente attraccano mercantili inviati dallo Spectrum. È ovviamente possibile in ogni momento avere la situazione del nostro esercito per sapere dove i rifornimenti sono più urgenti, così come possiamo conoscere la situazione logistica di ogni singola unità; per ottenere il massimo risultato occorre però saper utilizzare i dati a propria disposizione per ottimizzare spostamenti e tempi di decisione — come dire avere attitudine al comando...

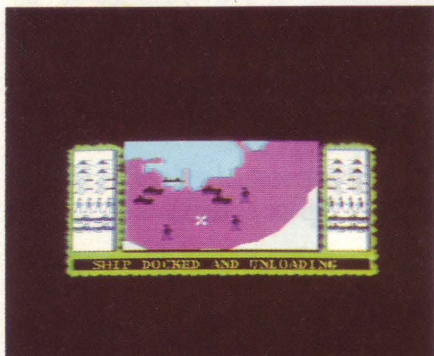
L'unico limite (relativo) di questo gioco è la velocità di svolgimento: ma è già sorprendente vedere una "scatoletta" come lo Spectrum gestire un programma del genere! Comunque, dopo un paio di minuti al massimo (in cui, di solito, tutti spostano le proprie armate con presunzione quasi hitleriana e piani a lunghissimo respiro) vi sono già decisioni cruciali da prendere, la situazione rifornimenti si fa assillante. In mancanza di una strategia efficace, si finirà per assistere senza possibilità di reazione alla distruzione totale delle nostre unità, impotenti di fronte alla supremazia avversaria. Per quando poi si saranno affinate strategie e tattiche, il programma saprà ancora metterci in difficoltà, grazie a un secondo livello di gioco, naturalmente più impegnativo. In conclusione, per chi ama i giochi





intelligenti (o è stufo del solito spara-e-fuggi e desidera provare qualcosa di più stimolante), ci troviamo di fronte a un gioco che vale senz'altro la pena di provare.







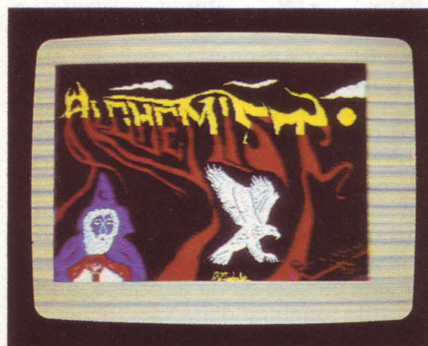
Alchemist

Una avventura sotterranea dalla mille sorprese

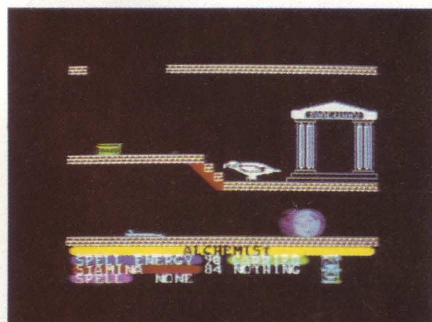
Si potrebbe definire una adventure animata. Siamo nelle vesti di un mago, capace di trasformarsi in aquila all'occorrenza, ed esploriamo le camere del

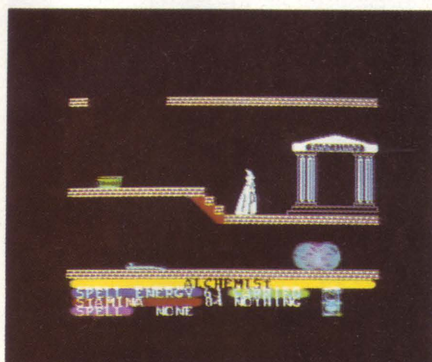


sotterraneo all'interno del quale ordisce le sue perfide trame il maligno Warlock. Il nostro compito è quello di trovare le quattro parti di un potentissimo incantesimo di distruzione e di utilizzarlo contro il nostro mortale nemico. Nelle varie camere del sotterraneo troveremo incantesimi di minore potenza ma indispensabili, armi, oggetti vari (spesso solo apparentemente fini a sè stessi, ma... non possiamo dire di più!) trappole, nutrimento e... le guardie di Warlock; massi rotanti, formiche, nubi magiche, teschi volanti — ed è solo l'inizio. Anche un mago, però, può avere dei problemi: questi sono costituiti dalla forza magica (che cresce col passare del tempo, ma diminuisce con l'uso) e la forza fisica (che decresce costantemente e ci costringe a cercare nutrimento... o altro...). Inoltre, non è possibile portare con sè nello stesso momento più di un incantesimo e un oggetto — o un'arma.



La grafica è perfetta e il gioco, fin dalla (bellissima) sigla iniziale contiene tutti gli ingredienti per avvincere fin dall'inizio e creare seri grattacapi per trovare... la fine! Dimenticavamo: il gioco necessita di 48K RAM.



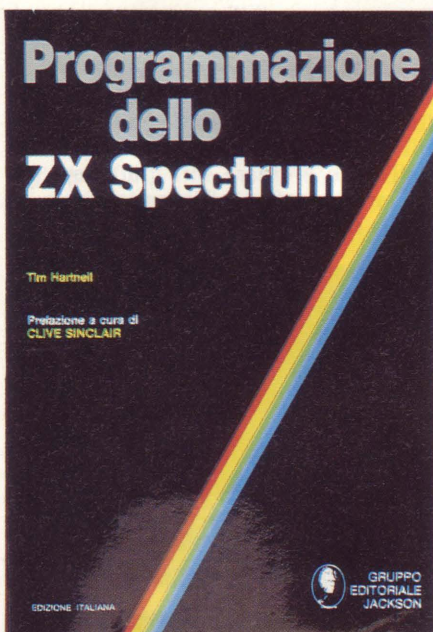


ALCHEMIST (cod. DIGMS03) e STONKERS (cod. DIGMS06) sono in vendita presso: TECHNOCLUB via Rosellini, 12 20124 Milano a L. 13.000 cad. più L. 2.000 per spese di spedizione. I programmi possono essere ordinati utilizzando il coupon pubblicato a pag. 66.

PRINT

Recensioni Libri

a cura di
Lucio Bragagnolo

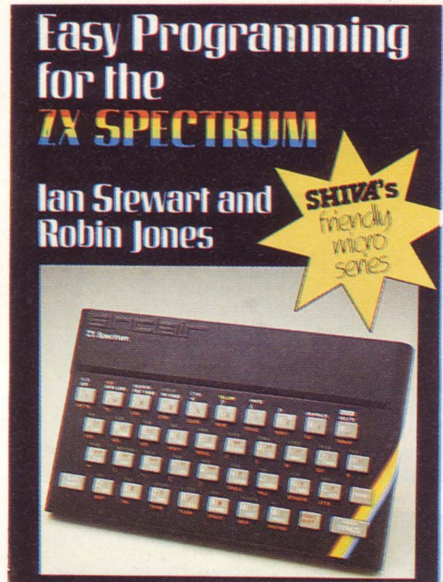


Programmazione dello ZX Spectrum

Tim Hartnell

prefazione a cura di Clive Sinclair
ed. Gruppo Editoriale Jackson
L. 18.000

Preso in mano il libro, la prima cosa che ha suscitato la nostra curiosità è stata ovviamente la prefazione, di Clive Sinclair in persona; ci chiedevamo che cosa avesse spinto il creatore dello Spectrum ad appoggiare con il suo nome questo testo, apparentemente uno degli ormai tanti in circolazione e, soprattutto, cosa avesse da dirci. Dobbiamo confessare che, dopo aver ritenuto il suo breve scritto un po' sbilanciato (vedi affermazioni tipo "Questo libro... vi aiuta a raggiungere una abilità di programmazione che, senza il suo aiuto, potrebbe richiedere anni per essere acquisita"), in seguito alla lettura ci siamo dovuti ricredere. Effettivamente sir Sinclair ha ragione: questo non è un libro come gli altri, o per lo meno non è come gli altri libri, bensì trattasi di — citiamo testualmente — "un utile complemento al manuale fornito con lo Spectrum", in grado di rivolgersi al programmatore in erba come a sinclairisti esperti che desiderassero aumentare le loro conoscenze.
Basta, per convincersene, dare un'occhiata al sommario: accanto ad argomenti "normali" (Esploriamo il colore dello Spectrum, Giocare con lo Spectrum), si possono vedere trattati temi piuttosto insoliti come "Usare lo Spectrum in affari", "Usare lo Spectrum a scuola" o "Grafica tridimensionale", capitoli in genere riservati a pubblicazioni specifiche. Evidentemente, non era possibile parlare diffusamente di tutte le applicazioni possibili di un calcolatore, per cui l'autore (il noto Tim Hartnell) ha dovuto forzatamente effettuare una scelta di argomenti e lavorare di sintesi: il tutto, però, senza rinunciare ad un approccio estremamente rigoroso con le materie trattate e ad una chiarezza e linearità di linguaggio esemplari.
In ogni capitolo si parte da considerazioni di carattere generale e alla portata del neofita più ingenuo, per arrivare progressivamente alla stesura di uno o più programmi inerenti al tema. Questi ultimi vengono dapprima scritti nella maniera più sintetica possibile, per poi venire arricchiti passo dopo passo da messaggi di input, opzioni particolari, commenti dettagliati e tutte le possibilità grafiche e sonore a disposizione dello Spectrum. Il lettore si trova così a disporre di una serie di programmi — da grafici a didattici, dai giochi alla piccola contabilità — di buona efficacia e semplicità, spesso collegabili con minima fatica ottenendo risultati a volte davvero sorprendenti: valgono per tutti come esempi il capitolo "affari", dove si arriva alla fine disponendo di un potenziale "pico-multiplan" domestico, o le pagine dedicate al sonoro, dove potreste avere qualche piccola inaspettata sorpresa, o la trattazione dell'istruzione INK 9 nella parte grafica ... troverete, alla fine, dopo il capitolo dedicato al linguaggio macchina, perfino una piccola storia dei calcolatori e una utile tabellina delle specifiche dello Spectrum.
Insomma, un libro, anzi, un complemento al manuale veramente alla portata di tutti e che fornisce la possibilità di compiere un salto di qualità come programmatore... o forse, in omaggio all'ultimo nato di casa Sinclair, dovremmo dire un Quantum Leap!



Easy programming for the ZX Spectrum (programmazione facile per lo Spectrum)

Ian Stewart/Robin Jones

ed.Shiva Publishing ltd.
prezzo medio di vendita L. 16.900

Nell'introduzione gli autori precisano come scopo di questo libro il "descrivere le possibilità dello Spectrum che un neoutente dovrebbe conoscere". E, sebbene questo sia solamente il primo di una serie di tre volumi (completata da "Further Programming for the ZX Spectrum" — programmazione avanzata per lo Spectrum — e "Spectrum Machine Code", il linguaggio macchina dello Spectrum) il materiale messo a disposizione dei lettori supera ugualmente di gran lunga i semplici bisogni primari degli acquirenti dell'ultima ora per soddisfare anche i programmatori medio-scarsi desiderosi di progredire. Il libro infatti comprende ben 26 capitoli, dal ritmo serratissimo, più una trentina di programmi pronti all'uso comprendenti il software gradualmente sviluppato negli esempi e alcune routine di varia utilità.

OFFERTISSIMA – EXELCO –

La EXELCO vi propone due vantaggiosissime combinazioni COMMODORE
Affrettatevi è un'OFFERTA irrepetibile!!!





Combinazione n° 1

n° 1 COMMODORE C64	L. 699.500
n° 1 Registratore «MAXTRON»	L. 82.000
n° 1 Libro «C64 EXPOSED»	L. 24.000
n° 1 Libro «Impariamo ad usare il personal computer	L. 25.000
n° 1 Cassetta Software «Garden Wars» della JCE	L. 20.000
Totale	L. ~~850.000~~

A SOLE L. 750.000
IVA INCLUSA

Combinazione n° 2

n° 1 COMMODORE VIC 20	L. 149.000
n° 1 JOYSTIC «TRIGA»	L. 31.000
n° 1 Libro «Impariamo ad usare il personal computer	L. 25.000
n° 1 Cassetta Software «ADVENTURELAND»	L. 49.000
n° 1 Cassetta Software «RIDE ON FORT KNOX»	L. 49.000
Totale	L. ~~303.000~~

A SOLE L. 258.000
IVA INCLUSA

Descrizione	Q.tà	Prezzo unitario	Prezzo Totale
COMBINAZIONE 1: n° 1 COMMODORE C64; n° 1 Registratore «MAXTRON»; n° 1 Libro «C64 EXPOSED»; n° 1 Libro «Impariamo ad usare il personal computer»; n° 1 Cassetta Software «Garden Wars» JCE		L. 750.000	
COMBINAZIONE 2: n° 1 COMMODRE VIC 20; n° 1 JOYSTIC «TRIGA"; n° 1 Libro «Impariamo ad usare il personal computer»; n° 1 Cassetta Software «ADVENTURELAND»; n° 1 Cassetta Software «RIDE ON FORT KNOX»		L. 258.000	

Desidero ricevere il materiale indicato nella tabella, a mezzo pacco postale contro assegno, al seguente indirizzo:

Nome

Cognome

Via

Città

Data C.A.P.

Desidero ricevere la fattura SI NO

Partita I.V.A. o, per i privati Codice Fiscale

PAGAMENTO:
A) Anticipato, mediante assegno circolare o vaglia postale per l'importo totale dell'ordinazione.
B) Contro assegno, in questo caso, è indispensabile versare l'acconto di Lire 50.000 mediante assegno circolare o vaglia postale. Il saldo sarà regolato contro assegno.
AGGIUNGERE: L. 5.000 per contributo fisso. **I prezzi sono comprensivi di I.V.A.**

EXELCO Via G. Verdi, 23/25
20095 - **CUSANO MILANINO** - Milano

SI ACCETTANO FOTOCOPIE DI QUESTO MODULO D'ORDINE

NON PERDETE QUESTE OCCASIONI!!!

Recensioni Libri

Sono interessanti, tra le altre cose, ben 5 capitoli dedicati all'impostazione ed esecuzione di un corretto e funzionale debugging, che potranno risultare molto utili specie agli autodidatti tutto genio e sregolatezza, sempre in difficoltà quando si tratta di "ripulire" e dare organicità alle loro (in genere) disordinatissime anche se spesso brillantissime creazioni. Quanto poi alle "chicche", segnaliamo il capitolo "Plotting" dedicato in gran parte al filling di figure chiuse — soprattutto circolari, le più ostiche — in grado di far risparmiare discrete quantità di tempo nel tentativo di risolvere il problema, "Curve plotting", un'esauriente e finalmente chiara guida alla rappresentazione grafica di funzioni (contenente tra l'altro i suggerimenti per modificare la scala di rappresentazione e un ampio set di funzioni già pronte per chi odia la trigonometria) e il capitolo "Peek and Poke", di introduzione ai primi rudimenti del linguaggio macchina. Gli altri capitoli, chi più chi meno, trattano le varie feature dello Spectrum con un impostazione di tipo manualista, metodica nell'esposizione e facente largo uso di esercizi ed esempi. Anche se in lingua originale, la comprensione del tutto non risulta particolarmente difficile, sia per l'inglese semplice utilizzato sia per la chiarezza di spiegazioni ed esempi. A volte la trattazione assume un ritmo piuttosto spedito che, gli autori hanno cercato di spezzare cercando di scrivere in uno stile colloquiale brillante e inserendo qua e là diverse vignette (queste, ahimé, un pò meno brillanti).
Anche se alcuni argomenti non vengono trattati o sono solo accennati (funzioni trigonometriche, definizione di funzioni proprie, INVERSE VIDEO e TRUE VIDEO, l'istruzione USR e altri, cosa del resto sinceramente ammessa dagli autori stessi), si potrebbe considerare questo libro come una "summa" per il programmatore agli inizi. Per i già decisi all'acquisto, un piccolo consiglio: non lasciatevi portare in un attimo alla fine, trascinati dai brevissimi capitoli e dal gran numero di argomenti, ma leggete lentamente, meditandoci sopra ed esercitandovi con calma e senza fretta sul vostro Sinclair, seguendo esempi ed esercizi; solo così facendo potrete far rendere al massimo le lirette che avete speso.

I libri recensiti possono essere ordinati utilizzando il coupon a pag. 66

Nel prossimo numero



ZX81 / Spectrum add-on: come e dove procurarsi ogni genere di accessorio per il vostro Sinclair



Le interruzioni, parte seconda
Nei meandri dello Spectrum: un'esplorazione all'ultimo bit



Cubo 2D • Bollettino vendite • Occhio alla frana • Attacco nucleare • (i vostri programmi!)



I contributi dei Club Sinclair

...e altro ancora!
Non perdete il numero 2 di SUPERSINC

ZX CLUB

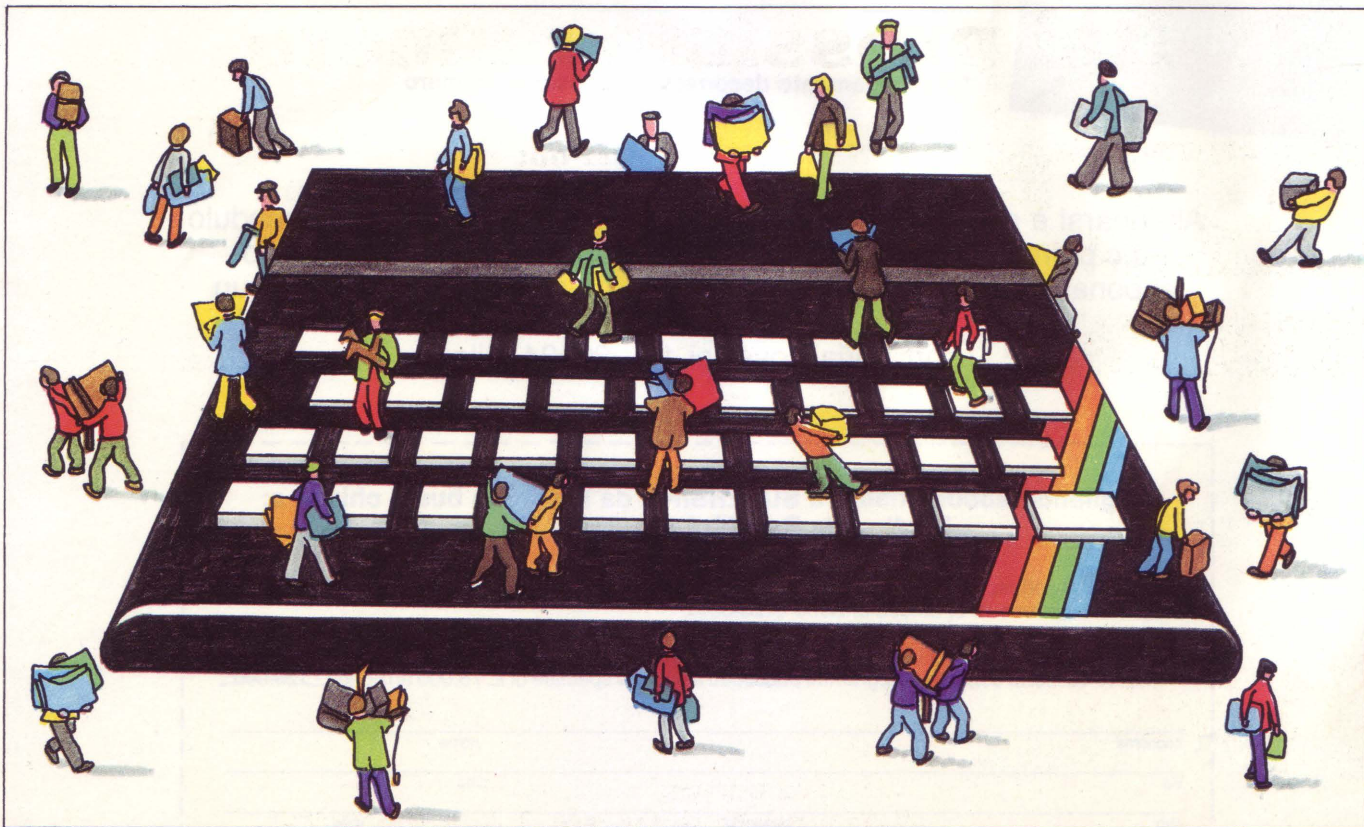
I Club Sinclair

Non appena passata l'onda di entusiasmo susseguente all'acquisto del calcolatore per tanto tempo agognato e terminata la fase di frenetico furore crativo con cui il neofita affronta i primi passi della sua avventura di utente di computer, subentra in genere la fase del "collegamento": ovverosia, una volta presa conoscenza dei propri, si spera provvisori, limiti di programmazione — e completato il saccheggio del software in vendita nel negozio sotto casa — si avverte l'esigenza di poter parlare con qualcuno che ne capisca un po' di più (e che conosca magari rivenditori più forniti e meno cari...). Purtroppo nella maggior parte dei casi gli amici propri o dei figli esauriscono assai in fretta le loro capacità di consulenza, rendendo necessario il rivolgersi a sconosciuti, con tutte le incognite del caso. Fortunatamente sembra essere una caratteristica tipica dei sinclairisti quello di essere maggiormente disposti, rispetto agli utenti di altre marche, a realizzare collegamenti riunendosi in associazioni e gruppi più o meno informali per trovare "amici di tastiera" (come direbbe Charlie Brown) con cui scambiare programmi, esperienze e notizie attraverso canali che spesso finiscono per far nascere autentiche amicizie anche fuori dall'ambito strettamente informatico.

La forma più appariscente — e concreta — di questo spirito di apertura verso l'esterno è sicuramente il Sinclair Club, tipo di organizzazione che sotto le più svariate denominazioni si occupa di mantenere i contatti tra gli aderenti, sviluppare software e hardware per tutti gli usi, acquistare e mettere a disposizione dei soci materiale spesso a prezzi scontati e sviluppare iniziative di tutti i generi secondo le proprie idee e possibilità costituendo un punto di riferimento insostituibile, specie in zone carenti di rivenditori aggiornati e di informazione.

Cosciente dell'importanza di questo fenomeno, SUPERSINC intende riservare ogni mese queste pagine a tutti i club sparsi per l'Italia pubblicandone notizie e iniziative, programmi e progetti hardware, ogni cosa che possa risultare interessante e meritevole di menzione. A tale scopo chiediamo la collaborazione dei club di ogni ordine e grado: mettetevi in contatto con SUPERSINC e forniteci notiziari, bollettini e ogni altro segno della vostra attività; saremo felicissimi di pubblicarlo (ovviamente dietro vostra autorizzazione).

Appuntamento quindi a tutti sulle pagine **ZX CLUB**!



NON UN ABBONAMENTO!

UN SUPERABBONAMENTO

Questo è ciò che la J.*soft* ti offre: un superabbonamento ad una super-rivista! Una eccezionale offerta di lancio: 16 numeri di SUPERSINC al prezzo di 10 oltre alla possibilità esclusiva di ricevere mensilmente, con la copia della rivista, il supporto magnetico con tutti i programmi pubblicati. L'offerta è valida fino al 30/6/1984.



LE NOSTRE SUPERPROPOSTE ABBONAMENTI

Abbonamento* a 16 numeri (5 nel 1984 + 11 nel 1985) **di SUPERSINC** al prezzo speciale di L. 35.000 anziché L. ~~56.000~~

Abbonamento* a 16 numeri di SUPERSINC + 16 cassette con tutti i programmi pubblicati al prezzo speciale di L. 75.000, anziché L. ~~120.000~~

***L'abbonamento decorrerà dal prossimo numero**

Abbonarsi è semplice! Effettuate il versamento con l'apposito modulo c.c.p. inserito in questo fascicolo, oppure ritagliate il tagliando abbonamenti pubblicato in questa pagina e speditelo allegando un assegno intestato a:
J.*soft* - Via Rosellini 12 - 20124 Milano.

Tagliando abbonamento a SUPERSINC da inviare in busta chiusa a:
J.*soft* - Via Rosellini 12 - 20124 Milano

☐ Desidero sottoscrivere un abbonamento a 16 numeri (5 nel 1984 + 11 nel 1985) di SUPERSINC al prezzo speciale di L. 35.000 anziché L. ~~56.000.~~

☐ Desidero sottoscrivere un abbonamento a 16 numeri (5 nel 1984 + 11 nel 1985) di SUPERSINC + 16 cassette con tutti i programmi pubblicati al prezzo speciale di L. 75.000 anziché L. ~~120.000~~

cognome ____________________ nome ____________________

via ____________________ città ____________________

cap. __________ provincia __________ data __________

firma ____________________

input·output

Piccoli annunci

Vendo Sinclair ZX 81 completo di alimentatore cavetti con imballo originale e garanzia con cassetta di 30 programmi e 2 libri a Lire 120.000 trattabilissime. Vero affare!!
Francesco Vellutini - Via Bezzecca, 6 - 20135 Milano - Tel. 5485649.

Sinclair club Mantova compra vende scambia programmi ZX Spectrum in tutta Italia ed estero banca dati di oltre 700 programmi per qualsiasi informazione.
Luca Gandellini - Via strada Cisa, 95 - 46047 Bancole di Porto Mantovano (MN) - Tel. 0376/398220.

Cambio o vendo per ZX Spectrum 200 programmi da L. 3.000 a 5.000 senza distinzione: games, utility, linguaggi. Chiedere lista senza impegno o telefonare.
Gianfranco Posterli - Via L. Ariosto, 123 - 20099 Sesto S.G. - Tel. 02/2480163.

Costruisco generatore di caratteri per ZX 81 aggiunge 256 caratteri al tuo computer mi basta un disegno su carta quadrettata dei nuovi simboli che vuoi il prezzo va da L. 55.000 a L. 75.000.
Sergio Coraglia - Via Tagliamento, 8 - 10096 Rivoli (TO) - Tel. 011/9591904.

Cambio, vendo programmi per ZX 81 con 1,4,16,32,64K RAM di memoria disponibili molti giochi, utilites, software aziendale e l'user definable graphics. Per avere direttamente e gratis la lista dattiloscritta o telefonare a:
Maurizio Marini - Via Mincio, 57 - 52100 Arezzo.

Scambio software per ZX 81 su listati o cassette inviatemi vostra lista che io invierò la mia.
Paolo Paldo - Via Circonvallazione, 30 - 20040 Cornate D'Adda - Tel. 039/6926260.

Vendo, scambio od acquisto programmi per ZX Spectrum.
Cerco, stampante ZX Printer od altra, ad un prezzo modesto.
Cerco, libri per ZX Spectrum.
Marco Arzani - Via Giovanni XXIII, 14 - 29017 Fiorenzuola D'Arda (PC) - Tel. 0523/982401.

Vendo, cambio programmi per ZX Spectrum, scrivete chiedendomi lista di programmi a L. 5.000 cad.
Francesco Avallone - Via Nicola Sole, 4 - 85100 Potenza - Tel. 0973/20281.

Per Sinclair Spectrum disponendo di un notevole archivio software. Vendo a L. 10.000 cassetta con 5 giochi a scelta. Telefonare ore pasti. Scrivere.
Ivano Parbuono - Via A. di Cambio, 4 - 37138 Verona - Tel. 045/568649.

Cambio programmi per ZX Spectrum 16-48K. Telefonare
Roberto Polastro - C.so Tassoni, 81 - 10143 Torino - Tel. 011/751453.

Vendo o cambio programmi per Spectrum (ne possiedo un centinaio circa) 16K e 48K. Scrivere per scambio elenco. Max serietà.
Luca Prian - Cannaregio 1091/e - 30121 Venezia - Tel. 041/713438.

Per Spectrum cambio e vendo software. L. 5.000 al programma (min. 4 programmi) cassetta compresa. Sono a Milano quasi tutti i giorni. Telefonate anche solo per farvi spedire l'elenco dei programmi.
Livio Scandella - Via De Amicis, 3 - 27029 Vigevano (PV) - Tel. 0381/72955.

Vendo programmi (tutti LM) per ZX Spectrum. Ogni programma L. 5.000 una cassetta da 90' con 15 giochi L. 45.000. Telefonare (ore pasti) o richiedete lista a:
Alberto Bianchini - Via Val Maira, 45 - 20162 Milano - Tel. 02/6429360.

Vendo causa acquisto Spectrum pocket computer Casio PB 100 + espansione (1Kbyte RAM) OR-1 + int. registratore FP12 + mini stampante FA3, tutto sole L. 300.000. Tutto il materiale è nuovissimo usato solo 2 settimane, è in garanzia fino al 1985.
Luigi Cerabolini - Via Felice Cavallotti, 91 - 27011 Belgioioso (PV) - Tel. 0382/960816.

Cambio centinaia di programmi ZX Spectrum da 16 Kbyte 48 Kbyte.
Alessio Santangelo - Via Luca Ganrico - 00143 Roma - Tel. 06/5912226-5040151.

Cambio, vendo programmi per ZX Spectrum. Prezzi imbattibili (min. L. 2.000 max L. 4.000 esc. cassetta e/o eventuali manuali). Scrivetemi per il listino. Se volete scambiare, mandatemi il vostro.
Stefano Giannici - Viale Francia, 4 - 90146 Palermo - Tel. 500190.

Cambio software per Spectrum 16 o 48 Kbyte dispongo di ottimo materiale giochi e applicazioni come: FORTH logassemblatore ecc. Scivetemi e vi manderò la lista.
Alessandeo Augello - Viale Del Fante, 56 - 90146 Palermo - Tel. 501740.

Vendo varie cassette per Spectrum: mastermind, battaglia galattica, pacman e tanti altri a sole L. 5.000 cad.
Sergio Aprile - Via Molinello, 70 - 86039 Termoli.

Vendo o cambio più di 60 programmi per ZX Spectrum fra cui giochi di ottima risoluzione grafica (burgertime, pacman, frogger, hobbit, crazy kong, vu-calc, vu-file, vu-3D, Pascal, Fortran) a metà prezzo. Vendo inoltre Atari 2600 in ottime condizioni con 6 cassette a sole L. 250.000.
Carlo Brini - Via Marciano di Eraclea, 25 - 00124 Roma - Tel. 6094506.

Spectrum, vendo-cambio programmi, cerco contatti con club o singoli possessori dello Spectrum. Risponderò a tutti.
Mario Pacchiarotti - Via Tuscolana, 884 - 00174 Roma - Tel. 06/7612788.

Vendo ZX 81 + memopak 32 Kbyte + alimentatore, cavi collegamento, 2 manuali, 1 libro di programmi, una decina di cassette (scacchi, data base, statistica, fantasy game). L. 250.000 in tutto. Telefonare ore pasti a:
Giuseppe Marini - Viale Primavera, 25A/3 - 16148 Genova - Tel. 010/330990.

Cambio, vendo programmi per ZX Spectrum 16/48 Kbyte, vendo cassette con 1/2 programmi a L. 5.000/7.000.
Stefano Sansavini - Via dei Macci, 18 - 50100 Firenze - Tel. 055/652321.

Compro listati o cassette per ZX Spectrum o TI/99/4D riguardanti il programma di computisteria e ragioneria per il triennio dell'istituto tecnico commerciale.
Ivo Spadone - Via C. Colombo, 6 - 18039 Ventimiglia (MI).

Occasione per ZX81 e per ZX Spectrum. Vendo l'interfaccia programmabile per joystick dell'Agfa a L. 50.000 (trattabili). Teledonare nelle ore serali.
Giuseppe Bungaro - Viale dei Monfortani, 12 - 00135 Roma - Tel. 06/33856611.

Vendo programmi ZX Spectrum flipper - H. hedges - luna crabs - starclash - bugaboo - death chase - paintbox - catalogo a richiesta L. 1.000.
Stefano Nocilli - Via Tuscolana, 224 - 00181 Roma.

Vendo programmi per ZX Spectrum a L. 6.000 cad. due programmi L. 10.000. Per informazioni telefonare a:
Marco Turiello - Via Fiordalisi, 1 - 20146 Milano - Tel. 02/4335382 (ore pasti) 3284770 (dopo le ore 20).

Vendo Sinclair ZX 81 completo di alimentatore cavetti per registrare memoria da 16 Kbyte RAM e manuali in inglese e in italiano il tutto a L. 140.000. Telefonare ore pasti a:
Pietro Di Giacomo - Via dei Filosofi, 14 - 06100 Perugia - Tel. 075/31129.

Cerco possessori di ZX Spectrum per scambio programmi.
Marco Varazi - Via Pinturicchio, 10 - 05100 Terni - Tel. 420282.

Spectrum vendo programmi gioco e utilità risposta garantita invio elenco gratuitamente scrivere o telefonare ore cena a:
Antonio Sfriso - Via O. Salamone, 7 - 301173 Mestre (VE) - Tel. 041/972887.

Programmi su nastro, Spectrum: 20 + 20 prg. (3 nastri diversi). ZX 81:50 + 50 progr. 1Kbyte, 35 + 35 progr. 16Kbyte, 20 maxi prog. 16 Kbyte). Incredibile sono impazzito: ogni nastro L. 7.000 solo se soddisfatti dopo 10 giorni prova.
Richiedili a:
Bruno Del Medico - Via Torino, 72 - 04016 Sabaudia.

Cambio, vendo software Spectrum 16/48 Kbyte - buon prezzo - inviatemi vostro elenco per scambio, vi invierò il mio, oppure richiedete la mia lista per acquisti allegando L. 400 in francobolli.
Erminio Benzoni - Via Del Cipresso, 4 - 20050 Perledo (CO) - Tel. 0341/830026 (ore serali).

Cambio, vendo software per ZX Spectrum 16/48 Kbyte.
Oltre 50 tioli a carattere commerciale ed altrettanti originali. L. 4.000/L. 5.000 rispettivamente per 16 e 48 Kbyte.
Richiedere lista inviando L. 400 in bolli.
G. Franco Posterli - Via Ariosto, 123 - 20099 Sesto S.G. (MI) - Tel. 02/2480163.

Vendo 25 giochi originali inglesi su cassette per ZX Spectrum a L. 50.000 + spese (spedizione - eventuali fotocopie di manuali). Dispondo inoltre di software di ogni tipo. Chiedere lista contenente oltre 200 titoli a:
Nicola D'Alessandro - Via C. Alcide De Gasperi, 413/D - 70125 Bari - Tel. 412470.

Per la pubblicazione dei vostri annunci utilizzare il coupon a pag. 66.

SUPERSINC LIBRI

CEDOLA DI ORDINAZIONE · LIBRI
da compilare e spedire in busta chiusa a
TechnoClub · Via Rosellini, 12 · 20124 Milano Tel. 6888228

Ordino i seguenti libri per un importo totale di L. .... + L. 2.000
come contributo fisso per spese di spedizione

Cod. .... Cod. .... Cod. ....

Cod. .... Cod. .... Cod. ....

☐ Contanti allegati ☐ Assegno allegato n° ....

☐ Ho spedito l'importo a mezzo vaglia postale

☐ Ho versato l'importo sul ccp. n° 19445204 intestato a Technoclub · Milano

☐ Pagherò in contrassegno al postino al ricevimento dei volumi (valido solo per i soci in Italia)

PROGRAMMAZIONE DELLO ZX SPECTRUM — cod. AJACO28 L. 16.200
EASY PROGRAMMING FOR THE ZX SPECTRUM — cod. BSHT003 L. 16.900

Nome ....

Cognome ....

Via ....

Città .... C.A.P .... Prov. ....

Se richiesta fattura:
Cod. F. e P. Iva ....

Data ....

Firma ....

Per i soci residenti all'estero — pagamento anticipato (vaglia o versamento su ns. ccp)

SUPERSINC SOFTWARE

CEDOLA DI ORDINAZIONE · software
da compilare e spedire in busta chiusa a
TechnoClub · Via Rosellini, 12 · 20124 Milano Tel. 6888228

Ordino i seguenti libri per un importo totale di L. .... + L. 2.000
come contributo fisso per spese di spedizione

Cod. .... Cod. .... Cod. ....

Cod. .... Cod. .... Cod. ....

☐ Contanti allegati ☐ Assegno allegato n° ....

☐ Ho spedito l'importo a mezzo vaglia postale

☐ Ho versato l'importo sul ccp. n° 19445204 intestato a Technoclub · Milano

☐ Pagherò in contrassegno al postino al ricevimento dei volumi (valido solo per i soci in Italia)

ALCHEMIST: — cod. DIGMS03 L. 13.000
STONKERS: — cod. DIGMS06 L. 13.000

Nome ....

Cognome ....

Via ....

Città .... C.A.P .... Prov. ....

Se richiesta fattura:
Cod. F. e P. Iva ....

Data ....

Firma ....

Per i soci residenti all'estero — pagamento anticipato (vaglia o versamento su ns. ccp)

SUPERSINC INPUT/OUTPUT

La rubrica INPUT/OUTPUT è gratuita ed aperta a tutti i lettori. Chi desidera comprare, vendere o cambiare hardware o software può inviare il tagliando a J.soft · Via Rosellini, 12 · 20124 MILANO

☐ COMPRO ☐ VENDO ☐ CAMBIO

☐ ZX80 ☐ ZX81 ☐ ZX Spectrum ☐ PERIF. ☐ SOFTWARE

....

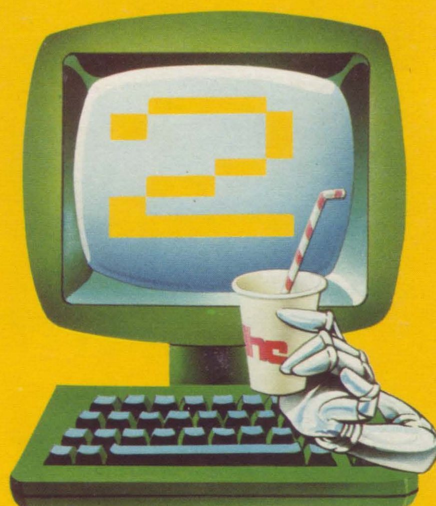
Nome .... Cognome ....
Via .... C.A.P ....
Città .... Tel. ....

SUPERSINC è bello, però... (ovvero suggerimenti, idee, critiche, richieste e tutto ciò che vi passa per la testa).

....

Nome .... Cognome ....
Via .... C.A.P ....
Città .... Tel. ....

HOME COMPUTER hc



LA RIVISTA DEL COMPUTER IN CASA

GIUGNO 1984 L. 3.500



UNA PUBBLICAZIONE
DEL GRUPPO EDITORIALE JACKSON

È IN EDICOLA
IL NUOVO NUMERO

TUTTE LE NOVITÀ DI LAS VEGAS
I SEGRETI DELL'AQUARIUS
A TU PER TU CON LA PROGRAMMAZIONE
SPECIALE GRAFICA

ZX 81 • VIC 20 • SPECTRUM • CBM 64 • ATARI • APPLE • SHARP

La prima ... ... l'unica

Enciclopedia di Elettronica e Informatica

Oggi con la 2ª Edizione ...la più aggiornata! In edicola.



Il successo si ripete

E.I. un prezioso strumento di formazione e di aggiornamento
a cui sono abbonati anche migliaia di specialisti, tra cui 4000 quadri FIAT

un orgoglioso primato dell'editoria italiana
alla cui pubblicazione sono interessati editori francesi, tedeschi, svedesi, canadesi, inglesi, sudafricani, portoghesi, spagnoli, australiani, zelandesi, messicani, sudamericani

uno strepitoso successo di lettori:
fino a oggi 6.000.000 di fascicoli venduti

una splendida opera da biblioteca
da 60 fascicoli settimanali
7 volumi - 1680 pagine - 700 foto
2200 illustrazioni a colori

E.I. una prestigiosa collaborazione tra
Learning Center
TEXAS INSTRUMENTS





GRUPPO EDITORIALE JACKSON